PATRICK RIEHMANN

ADVANCED VISUAL INTERFACES FOR INFORMED DECISION-MAKING

ADVANCED VISUAL INTERFACES FOR INFORMED DECISION-MAKING

PATRICK RIEHMANN

VIRTUAL REALITY AND VISUALIZATION RESEARCH GROUP
FAKULTÄT MEDIEN
BAUHAUS-UNIVERSITÄT WEIMAR

MAI 2015

SUPERVISOR & FIRST REVIEWER:

Prof. Dr. Bernd Fröhlich
Virtual Reality and Visualization Research Group
Fakultät Medien
Bauhaus-Universität Weimar

SECOND REVIEWER:

Assist.-Prof. Dr. Marc Streit
Institute of Computer Graphics
Johannes Kepler University Linz

Patrick Riehmann: *Advanced Visual Interfaces for Informed Decision-Making*, Gotha, Germany, Mai 2015,

ABSTRACT

This thesis presents new interactive visualization techniques and systems intended to support users with real-world decisions such as selecting a product from a large variety of similar offerings, finding appropriate wording as a non-native speaker, and assessing an alleged case of plagiarism.

The Product Explorer is a significantly improved interactive Parallel Coordinates display for facilitating the product selection process in cases where many attributes and numerous alternatives have to be considered. A novel visual representation for categorical and ordered data with only few occurring values, the so-called extended areas, in combination with cubic curves for connecting the parallel axes, are crucial for providing an effective overview of the entire dataset and to facilitate the tracing of individual products. The visual query interface supports users in quickly narrowing down the product search to a small subset or even a single product. The scalability of the approach towards a large number of attributes and products is enhanced by the possibility of setting some constraints on final attributes and, therefore, reducing the number of considered attributes and data items. Furthermore, an attribute repository allows users to focus on the most important attributes at first and to bring in additional criteria for product selection later in the decision process. A user study confirmed that the Product Explorer is indeed an excellent tool for its intended purpose for casual users.

The Wordgraph is a layered graph visualization for the interactive exploration of search results for complex keywords-in-context queries. The system relies on the Netspeak web service and is designed to support non-native speakers in finding customary phrases. Uncertainties about the commonness of phrases are expressed with the help of wildcard-based queries. The visualization presents the alternatives for the wildcards in a multi-column layout: one column per wildcard with the other query fragments in between. The Wordgraph visualization displays the sorted results for all wildcards at once by appropriately arranging the words of each column. A user study confirmed that this is a significant advantage over simple textual result lists. Furthermore, visual interfaces to filter, navigate, and expand the graph allow interactive refinement and expansion of wildcard-containing queries.

Furthermore, this thesis presents an advanced visual analysis tool for assessing and presenting alleged cases of plagiarism and provides a three-level approach for exploring the so-called finding spots in their context. The overview shows the relationship of the entire suspicious document to the set of source documents. An intermediate glyph-based view reveals the structural and textual differences and similarities of a

set of finding spots and their corresponding source text fragments. Eventually, the actual fragments of the finding spot can be shown in a side-by-side view with a novel structured wrapping of both the source, as well as the suspicious text. The three different levels of detail are tied together by versatile navigation and selection operations. Reviews with plagiarism experts confirm that this tool can effectively support their workflow and provides a significant improvement over existing static visualizations for assessing and presenting plagiarism cases.

The three main contributions of this research have a lot in common aside from being carefully designed and scientifically grounded solutions to real-world decision problems. The first two visualizations facilitate the decision for a single possibility out of many alternatives, whereas the latter ones deal with text at varying levels of detail. All visual representations are clearly structured based on horizontal and vertical layers contained in a single view and they all employ edges for depicting the most important relationships between attributes, words, or different levels of detail. A detailed analysis considering the context of the established decision-making literature reveals that important steps of common decision models are well-supported by the three visualization systems presented in this thesis.

ZUSAMMENFASSUNG

Diese Arbeit präsentiert neue interaktive Visualisierungstechniken und -systeme, die Einzelpersonen oder Gruppen als Unterstützung in konkreten Entscheidungsprozessen dienen, wie beispielsweise eine passende Formulierung in einer fremden Sprache zu finden, ein Produkt basierend auf einer großen Anzahl an Eigenschaften auszuwählen oder einen Plagiatsverdachtsfall zu beurteilen.

Der Product Explorer ist ein interaktives Visualisierungssystem auf der Basis von erweiterten Parallelen Koordinaten, das den Produktauswahlprozess in Fällen erleichtert, in denen viele Attribute und zahlreiche Alternativen in Betracht gezogen werden müssen. Eine neuartige visuelle Darstellung für kategorische Daten, die so genannten „extended areas“, in Kombination mit kubischen Kurven zur Verbindung der parallelen Achsen sind von entscheidender Bedeutung, um einen sinnvollen Überblick über die gesamte Produktauswahl zu ermöglichen und die Identifikation individueller Produkte und deren Eigenschaften zu erleichtern. Die visuelle Schnittstelle erlaubt es den Nutzern, die Produktsuche sehr schnell auf eine kleine Teilmenge oder ein einzelnes Produkt einzugrenzen. Die Skalierbarkeit des Ansatzes wird durch die Möglichkeit Entscheidungen für einzelne Attribute zu finalisieren und damit aus der Auswahl zu entfernen, deutlich verbessert. Darüber hinaus ermöglicht ein Attribut-Repository den Benutzern sich zunächst auf die wichtigsten Attribute zu konzentrieren und zusätzliche Eigenschaften für die Produktauswahl später in den Entscheidungsprozess einzubringen. Eine Benutzerstudie bestätigt, dass der Product Explorer gegenüber einem klassischen Webshop-Interface den Produktauswahlprozess auch für gelegentliche Nutzer deutlich verbessert.

Der Wordgraph ist eine interaktive, graphische Visualisierungstechnik, die Nichtmuttersprachlern hilft, passende Formulierungen in englischer Sprache zu finden. Ein Nutzer kann mittels Platzhalter („wildcards“) seine sprachlichen Unsicherheiten als Anfrage an den Netspeak Web-Service ausdrücken. Die Visualisierung zeigt die Ergebnisse als geschichteten, gerichteten und horizontal orientierten Graph. Jede Schicht des Graphen enthält die Wörter, die für einen Platzhalter gefunden wurden oder einem literalen Textfragment aus der Anfrage. Die Häufigkeit der Wörter in einer Schicht wird durch eine entsprechende Sortierung und visuelle Attribute verdeutlicht. Die Kanten des Graphen verbinden Wörter verschiedener Schichten und stellen so auftretende Wortverbindungen dar. Die graphbasierte Übersichtsdarstellung kann gefiltert und erweitert werden, bis eine geeignete Formulierung gefunden ist. Eine Benutzerstudie bestätigt, dass die graphbasierte Darstellung gegenüber einfachen Ergebnislisten desto mehr Vorteile bietet je mehr Platzhalter eingesetzt werden.

Aufgrund der schwerwiegenden Konsequenzen ist die Beurteilung eines Plagiatsverdachtsfalles mit besonderer Sorgfalt durchzuführen. Um diesen heiklen Prozess effektiv zu unterstützen, bietet die neuartige Plagiatsvisualisierung drei verschiedene Abstraktionsebenen, die durch entsprechende Navigations- und Selektionsmethoden verbunden sind. Die Übersicht zeigt die Zuordnung der über ein ganzes Dokument verteilten, einzelnen Fundstellen zu den Quellen an, aus denen möglicherweise kopiert oder paraphrasiert wurde. Ikonographische Darstellungen, die sogenannten Difflines, zeigen die innere Struktur einzelner Fundstellen und deren Gemeinsamkeiten und Unterschiede zu den entsprechenden Quellen. Fundstelle und Quelle können zudem mit einem gemeinsamen Textumbruchsverfahren nebeneinander angeordnet und im Detail analysiert werden. Plagiatsexperten bestätigen, dass dieses Visualisierungswerkzeug ihre Arbeitsabläufe effektiv unterstützen kann und eine deutliche Verbesserung gegenüber existierenden statischen Visualisierungen für die Beurteilung und Präsentation von Plagiatsfällen darstellt.

Die drei zentralen Beiträge dieser Dissertation zeigen wichtige Gemeinsamkeiten, auch jenseits ihrer Neuartigkeit, der wissenschaftlichen Fundierung im Gebiet der Informationsvisualisierung und der Unterstützung bei relevanten Entscheidungsproblemen. Der Product Explorer und der Wordgraph erleichtern die Auswahl aus vielen Alternativen, jedoch basierend auf unterschiedlichen Kriterien. Der Wordgraph und die Plagiatsvisualisierung fokussieren auf Textdarstellungen auf unterschiedlichen Detailstufen. Alle drei Entwicklungen orientieren sich an ähnlichen ästhetischen Kriterien, wie der klaren Strukturierung durch eine geschichtete Anordnung der graphischen Elemente und der Nutzung von graphischen Verknüpfungen, um die wichtigsten Beziehungen innerhalb der Datensätze zu verdeutlichen. Zudem wurden alle Arbeiten durch Benutzerstudien oder Befragungen von Experten evaluiert und in ihrer Nutzbarkeit und Nützlichkeit bestätigt. Eine abschließende Einordnung der vorgestellten Visualisierungstechniken in etablierte Modelle der Entscheidungstheorie zeigt, dass sie zentrale Schritte in diesen Prozessen effektiv unterstützten können.

PUBLICATIONS

[a] Patrick Riehmann, Henning Gruendl, Martin Potthast, Martin Trenkmann, Benno Stein and Bernd Froehlich. *WORDGRAPH: Keyword-in-Context Visualization for NETSPEAK's Wildcard Search.* IEEE Transactions on Visualization and Computer Graphics, pp. 1411-1423, September 2012

[b] Patrick Riehmann, Henning Gruendl, Bernd Froehlich, Martin Potthast, Martin Trenkmann and Benno Stein. *The NETSPEAK WORDGRAPH: Visualizing Keywords in Context.* Proceedings of the 2011 IEEE Pacific Visualization Symposium, pp. 123-130, Hong Kong, March 2011 (*Best Paper Award*)

[c] Patrick Riehmann, Jens Opolka and Bernd Froehlich. *The Product Explorer: Decision Making with Ease.* Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI), Capri, Italy, pp. 423-432, May 2012

[d] Patrick Riehmann, Martin Potthast, Benno Stein and Bernd Froehlich *Visual Assessment of Alleged Plagiarism Cases.* COMPUTER GRAPHICS forum, Volume 34 (2015), Number 3, EuroVis 2015, Cagliari, Italy, May 2015

Ever tried. Ever failed. No matter. Try Again. Fail again. Fail better.

— Samuel Beckett

Don't ask . . . to know about the giggle loop is to become part of the giggle loop.

— Jeff Murdock (Coupling)

ACKNOWLEDGMENTS

First and foremost, I have to thank Professor Bernd Fröhlich for giving me the opportunity to begin and accomplish this work, for introducing me to the scientific work process, and for supporting me along the way.

Second, I want to thank Marc Streit for agreeing to be second reviewer on such short notice.

Furthermore, a particular "thank you" to all people who helped with the papers and this thesis: most notably, Henning Gründl, Jens Opolka, Martin Potthast, and Benno Stein.

Above all else, I want to thank my family: my wife Anja and my children Paula and Karla for enduring frequent bad moods and being agitated on many weekends before paper deadlines and before finishing this thesis. I'd especially like to thank my parents, Hubert and Beate, for their tireleless support in every possible way.

I would also like to thank people who influenced me, inspired me, or helped me build my skills.

CONTENTS

Part 1

THE DECISION CHALLENGE

1.1 IT'S YOUR CHOICE

At any age, human decision-making is based upon the information available. This cognitive procedure is not only pertinent to the so-called information age: what actually matters is the increase in both the amount and complexity of information to be considered before arriving at an informed choice. Seemingly related, searching the web for appropriate help in making decisions reveals to be a complex task, as *Amazon.com*[1] with over 405 000 results unveils. The 38 subtopics the results were categorized under express how decision-making influences all aspects of life; most notably *Business & Money*, *Management*, and *Politics & Social Sciences* provided a majority of hits, succeeded by *Science & Math* and *Education & Teaching* with almost 20%. Also of great interest is advice in *Reference*, *Personal Transformation & Self-Help*, and *Spirituality & Religion*. However, titles like "Food Logic: Making Smart Decisions for Your Dog in an Age of Too Many Choices" [11] or "Choosing Kitty: Making Decisions About Kitty Caring" [21] indicate that, even for hobbies and leisure activities, a strong need exists for being guided in making informed decisions on such matters.

Despite the commercial aim of an internet book selling site, the many different categories resemble the various research fields that are committed to the process of decision-making, although neither in the same order nor distribution as a look at *Microsoft Academic Search*[2], with over 312 000 results, reveals. In science *Computer Science* followed by *Medicine* outnumber *Economics & Business* and *Social Sciences* by more than double in paper count. This is, at first, a surprising, yet comprehensible, result regarding the increased computer-based algorithms that are needed to face the so-called *information overload* when trying to make a decision. This presents an ironic situation since the overwhelming data is more and more caused, tracked, and captured by algorithms introduced by computer experts, which yields a self-amplificating loop multiplying the data at each iteration. Particularly in medicine, the data available on the current status of a patient has grown exponentially during the last 20 years (especially since medical imaging has become increasingly important) [48].

Numerous comprehensive approaches, more or less similar, try to define the term *Decision-Making*. Following, for example, Wang and Ruhe [121] it "is a process that chooses a preferred option or a course of actions from among a set of alternatives on the basis of given criteria or strategies." This introduction is not intended to argue whether human decisions are rational, as considered by Edwards [22], or emotionally influenced and far from being ideal, nor will we dive into the differences of the normative or the descriptive paradigm. Just for starters, the normative approach is more interested in how a decision should be made, while this thesis focuses more on

1 Accessed Jan. 5th, 2015
2 Accessed Jan. 6th, 2015

supporting actual decisions and, therefore, on the descriptive paradigm that attempts to explain how decisions are really made by humans.

Although not always presented in these exact words, supporting decisions is, ultimately, a core application of Information Visualization and Visual Analytics. This proposition is fostered by scrutinizing the most important answers "any technology that claims to overcome the information overload problem has to provide" (Keim et al. [54], p155), whereas the latter three of these four questions are related to decision-making: (1) "Who or what defines the relevance of information for a given task?" (2) "How can appropriate procedures in a complex decision making process be identified?" (3) "How can the resulting information be presented in decision- or task-oriented ways?" (4) "What kinds of interaction can facilitate problem solving and decision making?"

Nevertheless, there are many cases of visualizing data retrospectively for information gain only or just for enjoyment as also mentioned by Munzner [73] (page 68; enjoy goal) to "refer to casual encounters with vis". Her example in this regard, the Name Voyager [122], "a vis tool originally intended for parents focused deciding on what to name their expected baby, ended up ..." being used to analyze historical trends for the sake of entertainment only. Another example, not really related to entertainment, is the visualization of the Titanic dataset used for introducing Parallel Sets [59].

As mentioned in the beginning, the most important field related to decision theory is business processing. Visualization of business information plays a more and more important role according to the increasing number of whitepapers, guidelines (e. g. TWDI [105] , SAS [45]) or articles such as "The rise of the dataviz expert" [40]. It shows the changed awareness of the matter in the age of big data, especially, for supporting decision-making in business administration by utilizing appropriate visualization. Sometimes, the impression is that visualization is a novel topic. Yet, there is a long tradition of business decisions being supported by some kind of visual presentation of information. For example, in the 1990s, such applications were labeled Decision Support Systems (DSS) and were often more promising than useful [86]. This term is still common in some business areas. It appears that the topic has been concealed for more than a decade, especially in business areas, before receiving more and more attention by the time the amount of data swelled in the wake of modern search engines, WEB 2.0, and social media, as well as the appearance of small and connected information-capturing devices such as various sensor types used in industrial domains, digital cameras, or smartphones, for example.

In this thesis, three advanced visual interfaces for facilitating complex decisions in different areas are introduced: the Product Explorer, the Wordgraph, and an interactive visualization for exploring and assessing cases of plagiarism.

First, the Product Explorer is intended to support people in making product decisions. It is an interactive visualization for choosing a certain product over numerous alternatives sharing the same attribute set. Parallel Coordinates are enhanced to appropriately depict quantitative information of categorical and ordered data with few occurring attribute values and to guide the user while expressing his/her requirements regarding the desired attribute variation and, therefore, eventually narrowing down to a fitting subset or even one remaining product. All products and important attributes are visible at a glance throughout the selection process and all interactions can be performed with immediate feedback, hence users do not have to wait until a query response appears. Generally, the Product Explorer is able to cope with any kind of multivariate decision, either personal or business-related

Second, the Wordgraph provides an example for supporting people with limited experience in a particular matter, which is why they are often uncertain about the specific details. Non-native speakers can address these issues by integrating wildcards into a phrase they are uncertain about. The graphical response visually suggests appropriate word sequences and interactively guides the user in choosing one of them. It can be used in various fields, including science, business, or education.

Third, the Plagiarism Visualization supports superiors in considering and presenting alleged cases of plagiarism in works such as PhD, master, or bachelor theses. Nevertheless, due to the severe consequences of plagiarism, both morally and legally, a thorough investigation by the council in charge, as well as a presentation that backs the eventual verdict, has to be strived for if such an allegation has been made. A three-level approach guides the user in scanning and exploring all of the so-called finding spots, which are suspicious text fragments that have noticeable similarities to existing texts written by other authors. The overview depicts the locations of these finding spots in relation to their possible source documents. Each single finding spot is illustrated by an intermediate representation called a diffline, which provides a glyph-based overview of finding spots by visually encoding the differences and similarities of the two text fragments. For further investigation, the diffline can be expanded to a novel kind of text view. This side-by-side view wraps both texts, the suspicious one and the possible original, by aligning their equal parts vertically in a way that they serve as a skeleton for the parts that have been changed. Our prototype provides effective means to navigate and filter the finding spots of the entire case and enables direct interaction between finding spot, original, and their diffline.

However, what does all that mean in decision-making theory? The book published in 1910 *How to think* [19] by John Dewey is typically considered the origin of decision theory (albeit some older sources exist [39]). He described a decision as an iterative process of five steps: " (1) Feeling of a difficulty, (2) Defining the character of that difficulty, (3) Suggesting possible solutions, (4) Evaluating suggestion, and (5) Further observation and experiment leading to acceptance or rejection of the suggestion."

Five decades later, Simon and Brim (independently of one another) expanded on this model. Simon defined three steps led by his observations of organizational processes [101]: (1) Intelligence (in a military sense): A phase of more or less continuous scanning and recognizing that a decision has to be made due to events of changed status of the environment. (2) Design: Acquiring necessary information and designing a set of possible choices. (3) Choice: Assessing and choosing one of the detected alternatives.

On the contrary, Brim1962 [8] provided an approach with six sub-procedures: " (1) Identification of the problem, (2) Obtaining necessary information, (3) Production of possible solutions, (4) Evaluation of such solutions, (5) Selection of a strategy for performance and, (6) Implementation of the decision."

In subsequent years, criticism about the sequentiality of those models arose. Notably, Witte [132] stated that, due to the nature of human cognition, some of the defined steps are processed parallel or simultaneously rather than subsequently. Mintzberg, Raisinghani, and Théorêt [69] proposed a non-sequential model by enhancing Simon's work towards a cyclic approach that allows to return to a previous step at any time if the current step cannot be passed satisfactorily, as well as the possibility to start again. Nevertheless, I argue that loops within the models are not a real issue, to a certain extent, when they are covered by appropriate interaction that supports such non-linear scenarios.

A particular perspective on decision-making that should also be introduced is the so-called conditions under which a decision has to be made. Knight [56] described three conditions. (1) Certainty: which means that for each alternative the consequences are known. (2) Risk: each alternative can be followed by a set of consequences; also the probability of each consequence is known. (3) Uncertainty: contrary to risk the probability of every consequence is not known, each single consequence itself is.

Many methods and techniques were presented in several fields of information visualization for coping with the challenge of decision, though no particular survey seems to exist according to actual and real-world decision-making. Therefore, within the scope of this thesis, I would like to propose another classification more oriented toward concrete or actual decision tasks, as well as how people can be informed visually in making such a decision. Admittedly, this classification is derived from the application domains of my thesis and is certainly neither complete nor does it cover all decision problems.

DECISION-MAKING UNDER UNCERTAINTY Like the Wordgraph, it refers to applications that people support in overcoming gaps in their knowledge or weaknesses in their skill set at certain moments. In the context of writing in a foreign language, web-based systems are helpful starting from dictionaries (*dict.cc*) or thesauri (*openthesarus.de*) to more sophisticated tools like *Linguee.com* or *netspeak.cc*. See also Section 1.3 and 2.2. Please be aware that

the meaning of uncertainty differs from the meaning of same term in Knight's work.

MULTIVARIATE CHOICE Sometimes called Preferential Choice or, in other domains, denoted as Faceted Search. This contains systems that support decisions concerning multivariate data: for example, simulation data, manufacturing data and measured data. Besides time-based data, one of the topics that yielded the most publications in Information Visualization. Examples can be found in Section 1.3 and 3.2.

CONSIDERATION AND ASSESSMENT Coming to a verdict based on a thorough and detailed investigation of the matter. Popular examples are detecting tax evasion or inspecting money laundering. Additionally, time-critical problems like medical decisions or control room scenarios can be considered here as well (more in 1.3 and 4.3).

The work on the visualizations systems designed and created for this thesis was essentially motivated by a sense of dissatisfaction regarding the presentation of particular data sets. The common thread of all solutions, literally, was directly connecting the most important elements by visual links. For example, a long list consisting of many similar phrases (a result of an early Netspeak prototype) seemed to be too unstructured and too redundant for a clear and legible presentation. Hence, the vision of a connected structure appeared that emphasizes the words most commonly used, as well as their collocations in a visual manner. An approach that matured into the Wordgraph after many iterations.

Early ideas of Product Explorer were primarily driven by thoughts about the visual sculpting process described by Elmqvist in the paper "Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation" [24], which appeared to me as being far too complicated for this task. A Parallel Coordinates display enhanced for categorical or mixed data should suffice for providing a much easier and faster interaction process to select a product. Additionally, employing a concept similar to Sankey Diagrams [90] for expressing the quantities in the system was apparent from the beginning.

On the contrary, the design of the Plagiarism Visualization lasted longer and was more traditional in terms of a scientific process. This was due to an extensive phase of searching both related and established solutions, found in only a handful of tools, that provided some kind of overview or applied proper visualization and interaction techniques. Early prototypes aimed at presenting the entire suspicious documents in a scrollable view were discarded once the concept of the linked overview came up – maybe it was too obvious at first. Focusing on the individual finding spots to speed up the assessment process ultimately resulted in a need for a visual abstraction, yet it took a while to develop it into a glyph-based concept called a diffline.

1.2 OUTLINE

This thesis is structured as follows:

PART 1 surveys decision-making. It introduces different theoretical aspects and models, shows the relations regarding business administration, and explains the increasing importance of visualization in both fields. Furthermore, related work that supports decision-making by providing interactive visualization systems is discussed.

PART 2 presents the paper "Keyword-in-Context Visualization for Netspeak's Wildcard Search", which is the extended journal version for IEEE Transactions on Visualization and Computer Graphics (TVCG) of the best paper award winning publication "The NETSPEAK WORDGRAPH: Visualizing Keywords in Context" (PacificVis'11).

PART 3 consists of the paper "The Product Explorer: Making Purchase Decisions with Ease" and further unpublished material.

PART 4 presents an extended version of the paper "Visual Assessment of Alleged Plagiarism Cases."

PART 5 concludes with the contributions of this thesis, especially, in light of the described decision models. It discusses similar visual principles all three visualization systems are based on and gives an overview of ideas for future developments.

1.3 DECISION RELATED VISUALIZATIONS

This chapter addresses a selection of important visualization paradigms and visualization systems that focus on decision-making in domains mostly related to the work presented in this thesis, starting with papers related to decision-making under uncertainty, followed by work aimed at exploring multivariate data, and ending with visualizations intended to support consideration and assessment both with and without time constraints. Additional related work concerning more specific topics of the three interactive visualizations can be found in Sections 2.2, 3.2, and 4.3).

In Overview [7], Brehmer et al. provide an example of deciding under uncertainty with a visual system, particularly aimed at investigative journalists, for exploring large corpora, including law compilations or leaked documents, in order to choose material worthy for publication or to decide which texts back an ongoing story. Another system intended to draw conclusions within its specific field is SoccerStories, a "visualization interface to support analysts in exploring soccer data and communicating interesting insight" [79].

As already referred to, multivariate choice has been a challenge for more than two decades. Early papers about Dynamic Queries and The Dynamic Home Finder by Shneiderman [98] and Williamson [131] were suited for the task of finding real-estate property. They were one of the first examples of interactive visualizations that combined several sliders for querying with immediate responses shown on a map. More general tools for analyzing multi-attribute data were the Influence Explorer [112], as well as the Attribute Explorer [104], both providing interactively-linked histograms combined with range selections for narrowing down subsets that match the given requirements.

Later on, Bautista et al. [3] presented redesigned Value Charts (originally [10]) that tried to support all stages of preferential choice that had been concluded by them, which is why the paper is also substantial regarding theoretical aspects of this matter. A subsequent study examining "the Impact of User Characteristics and Different Layouts on an Interactive Visualization for Decision Making" by Conati et al. [17] revealed that tasks are affected by user characteristics and horizontal layout are preferred for "users with low visual working memory." Another study by Yi [136] tried to prove the usefulness of a visual decision tool for coping with information overload, particularly for the domain of choosing a nursing home.

A older, yet powerful, tool"for consumer-based information exploration and choice," aimed at buying cars, is Parallel Bargrams (or EZChooser) [133]. Each dimension is an equal-length horizontal bar consisting of clickable buttons for each value, whereas the button length corresponds to the number of products that share that value. Parallel Sets [59], already mentioned, is an interactive interface for exploring the relationship within a categorical data set (see further details in 3.2). The interface of Parallel Bargrams is reminiscent of the proportional elements of the layers/axes displayed by the Product Explorer and the Parallel Sets, but without the single paths that represent each individual product or ribbons passing from one category of one axis to another category on the next axis, respectively.

Another possibility for choosing products is described in the paper by Elmqvist et al. [24]. The individual scatterplots within the matrix are mapped on a rotating cube that can be used (like turning a dice) to navigate and, therefore, visually sculpt a subset of the product data (cameras in this case) that matches the user's wishes. More suited to analyzing and comparing groups of dimensions of biological data "in an arrangement of heat maps reminiscent of parallel coordinate" is the visualization by Lex et al. [61].

Icon-based visualizations are an appealing alternative for displaying multivariate data containing less attributes. The so-called Chernoff Faces [13] were introduced to take advantage of human sensitivity with respect to recognizing facial characteristics for visually expressing multivariate data. Different features, for example, eye size or nose length, were utilized to map different attributes of a data set. In the original approach, up to eighteen attributes were mapped, a number that is hard to believe

being reasonably applicable. The maps by Dorling [20], which survey the social and economical differences in the United Kingdom with only four different visual attributes, seem to be more convincing. Concerning multivariate purchase decisions, Spence and Parr [103] used miniature house icons (as direct metaphorical relation) to encode eight attributes of houses or real estate. Their study revealed significant time savings of such direct metaphorical icons over textual descriptions for choosing an appropriate estate. In my paper "Visualizing Food Ingredients for Children by Utilizing Glyph-Based Character" [92], the idea of directly related icons is applied to children between the ages of four and eight by introducing comic-like characters whose shape and features depend on the main ingredients of food products. The study showed that children are able to recognize several ingredient manifestations encoded as visual attributes.

Starplots and Radarplots might also be considered as some kind of icon-like depiction. A very early example from 1877 is described in the book by von Mayr [120] (as well as an early example of a Treemaps predecessor). Both glyphs are restricted according to the number of displayable attributes or dimensions. Albeit not a glyph-based technique per se, Andrews curves [1] are an interesting way to show multivariate data. Each attribute value of a single data set entry is used to create a Fourier series, which is subsequently plotted between a given range, resulting in a plot containing as many lines as data entries exist.

Compared to icon-based approaches, the Parallel Coordinates concept invented by Inselberg [113] is often more suited (at least theoretically) for showing larger number of attributes or dimensions. Looking at all attributes at once in a single view, and being able to interact with the data, provides an advantage with large data sets. Further readings about can be found in Section 3.2.

Besides Parallel Coordinates and its numerous descendants, plot matrices are often used as abstraction for choosing among multivariate data. Best known, of course, is the scatterplot matrix. Later approaches tried to extend the plot matrix concept to other applications, like Hyperslice by van Wijk et al. [117], to provide a new view of scalar multi-variable functions. With the"Generalized Pairs Plot," Emerson et al. [26] proposed "a generalization of the scatterplot matrix based on the recognition that most data sets include both categorical and quantitative information" (an observation that is not new regarding the development of the Product Explorer). In his approach, several kinds of plots are employed like mosaic plots, facetted bar charts, and fluctuation diagrams, depending on which kind of data are paired and the user's preferences. A package for the programming language *R*, which focuses on statistical computing, is available [25]. GPLOM [49] is an interactive variant of this proposal enriched by textual search, albeit restricting the plots only to scatterplots, heatmaps, and barcharts. The paper also contains an interesting study of the described prototype against the Tableau software [102], a widely-used and professional software focusing on Information Visualization and Visual Analytics.

Gratzl et al. presented LineUp [36] for interactively comparing multiple ranked data, an approach well-suited for almost any kind of decision based on multivariate data. The RankExplorer by Shi [97] followed a similar direction, but focused on time by extending the Theme River [42] to a "visualization of ranking changes in large time series data."

Beyond the typical mouse- or touch-based interfaces, Klum et al. introduced "Stackables: tangibles designed to support faceted information seeking" [55]. Each stackable is a hand-sized cube with a small screen which represents one facet or one attribute that is adjustable by two control wheels. Multiple stackables together constitute a tangible query formulation interface for product search, for example.

With Pargnostics, Dasgupta et al. [18] proposed a screen-space metric, particularly suited for Parallel Coordinates displays. Bertini [5] presented a more general "systematization of techniques that use quality metrics to help in the visual exploration of meaningful patterns in high-dimensional data." Another topic related is geographic information systems. The idea is to ease decisions by depiction results of multi-criteria analysis directly onto a map view or a spatial representation. Andrienko et al. [2], for example, "suggest several mechanisms for linking and coordinating visual exploratory" to be applied between different views during the choice phase of Simon's model.

No field is so sensitive concerning severe consequences of decisions as medicine; time is often the most important issue here. Therefore, numerous systems have been presented to support decision-making concerning a patient's condition or to optimize their further and future treatment, such as Lifelines [80], VisuExplore [93], and others. The paper by Rind et al. [94] reviews and compares the most important visual interfaces regarding that matter. More recently, Gotz et al. [34] published "DecisionFlow: Visual Analytics for High-Dimensional Temporal Event Sequence Data," which is also applicable for scrutinizing electronic patient records as a "scalable and dynamic temporal event data structure with interactive multi-view visualizations and ad hoc statistical analytics." VIE-VISU [47], however, differs from the already mentioned systems, because it tried to visually express multivariate vital parameters of patients in a hospital over time as multiple glyphs or, in their terms, as "metaphor graphics" A visualization for improving personalized medical treatment in tumor therapy was presented by Streit [106] which is intended to "efficiently compare multiple patient stratifications." Disaster management and prevention is also an important field concerning immediate decisions. RunWatchers by Konev et al. "is a simulation-based approach to design protection plans for flood events" by generating "large and complex decision trees," as well as "automatic storyboards to convey plan details and to justify the underlying decisions" [58].

"Although most transactions are legitimate," fraud, such as tax evasion, and money laundering, is a serious problem in the financial sector, especially considering the

increase in transactions during the recent decade. Chanh et al. [12] provides "coordinated views used to depict relationships among keywords and accounts over time" based on high-scalable database to "show similar transaction patterns." With BaobabView, Elzen et a. [115] tried to achieve a similar objective by using "interactive construction and analysis of decision trees." Malik et al. "present a visual analytics approach that provides decision makers with a proactive and predictive environment in order to assist them in making effective resource allocation and deployment decisions in Criminal, Traffic and Civil (CTC) incident datasets" [63].

Parallel Tag Clouds are an interactive system utilized to "explore faceted text corpora" and are specifically used to determine which court of appeal should one choose for a particular law case and which should be avoided by all means. Due to its visual combination of Tag Clouds and Parallel Coordinates, it can be considered as multivariate decision. Madhavan et al., however, presented a solid example for consideration and assessment with a system for enabling researchers to analyze "their research funding portfolio in order to make more informed decisions of their future funding" [62].

In order to conclude this section and to get back to where I started from in the introduction, we have seen that business and decision-making are strongly related and visualization systems play an increasingly important role for enterprise data analysts. Surprisingly, there is "little research on how analysis takes place within the social and organizational context of companies." Kandel [53] tried to fill this gap by conducting a series of semi-structured interviews with data analysts from organizations in different sectors.

Part 2

KEYWORD-IN-CONTEXT VISUALIZATION FOR NETSPEAK'S WILDCARD SEARCH

The Wordgraph helps writers in visually choosing phrases while writing a text. It checks for the commonness of phrases and allows for the retrieval of alternatives by means of wildcard queries. To support such queries, we implement a scalable retrieval engine, which returns high-quality results within milliseconds using a probabilistic retrieval strategy. The results are displayed as Wordgraph visualization or as a textual list. The graphical interface provides an effective means for interactive exploration of search results using filter techniques, query expansion and navigation. Our observations indicate that the textual interface is sufficient for the phrase verification, both interfaces support context-sensitive word choice, and the Wordgraph best supports the exploration of a phrase's context or the underlying corpus. Our user study confirms these observations and shows that Wordgraph is generally the preferred interface over the textual result list for queries containing multiple wildcards.

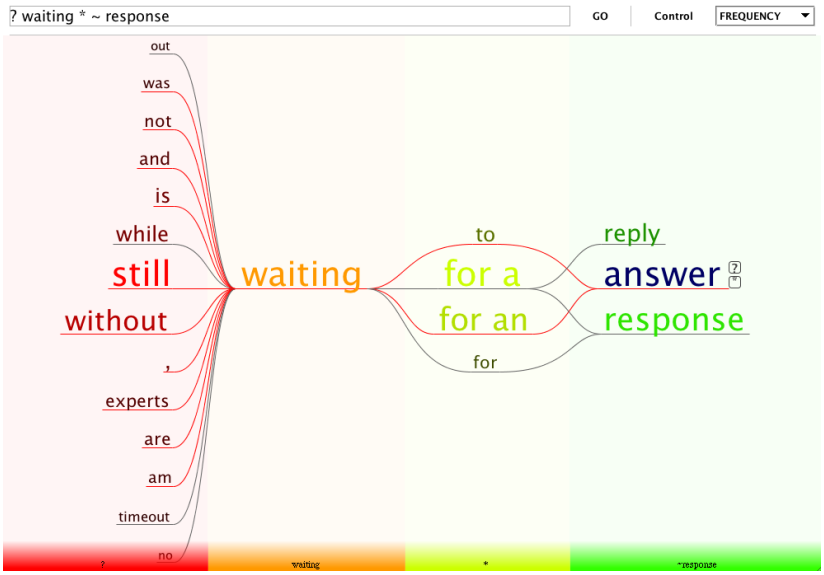
2.1 INTRODUCTION

The Wordgraph is a visual tool for context-sensitive word choice. It allows its users to check the commonness of a phrase and to express uncertainties in choosing words by formulating queries that contain wildcards. The search results are transformed to a legible and interactive graph visualization—the Wordgraph (see Figure 1). The graph layers follow the structure of a query, showing one layer for every literal word and wildcard, which are filled dynamically with the results obtained from Netspeak's retrieval engine. Search results visualized by the Wordgraph can be interactively explored, refined, and expanded by means of filter techniques and navigation. Queries are processed by a scalable retrieval engine called Netspeak, which returns high-quality results within milliseconds using a probabilistic retrieval strategy. It indexes a corpus of more than 3 billion word n-grams up to a length of $n \ 5$ words along with their occurrence frequencies in a large portion of the Web.

Wordgraph's intended audience face difficulties in this respect, since their innate sense of language—their sprachgefühl—is often not sufficiently developed. They might ask themselves how others would formulate a particular phrase; a piece of information that is generally hard to come by. Netspeak implements a statistical solution by contrasting alternative phrases based on their absolute and relative occurrence frequency. Our working hypothesis is that choosing more common phrases over uncommon ones may improve readability, comprehensibility, and writing style. Obviously, this is not true in general, but as non-native speakers we found Netspeak's suggestions immensely helpful in all our daily writing tasks.

A variety of visualizations of relations among words, phrases or collocations (also called "keywords in context") have appeared in recent years, such as the WORDTREE, PHRASENETS [116], Google Scribe, and AWKCHECKER, to name only a few. The WORDTREE [123] employs suffix trees to index text and to visualize the tree starting from the query word(s). PHRASENETS encode subject-predicate-object triplets in a directed graph, which are mined from a text by specifying the predicate and considering subject and object as wildcards (e.g. ? loves ?). Google Scribe [33] assists authors with writing by suggesting the next word in a phrase—very similar to AWKCHECKER [77]. Both systems are (likely to be) based on language model theory.

The Wordgraph visualization and the Netspeak engine can be considered as a generalization and a combination of the aforementioned approaches: the Wordgraph has more complex layout constraints than the tree layout of the WORDTREE. PHRASENETS visualize only triplets with a fixed predicate and the force-based layout limits the phrase legibility. Both tools focus on corpus exploration instead of being interactive word choice tools. AWKCHECKER and Google Scribe do not employ visualizations at all, as they are not intended for ad hoc wildcard queries.



(a)

? waiting * ~response Search

Frequency		Phrase 1 2	Example
9,668	13.6 %	still waiting for a reply	⊞
8,753	12.4 %	without waiting for an answer	⊞
7,904	11.2 %	still waiting for an answer	⊞
5,118	7.2 %	still waiting for a response	⊞
4,728	6.7 %	while waiting for a response	⊞
3,677	5.2 %	is waiting for a response	⊞
3,432	4.8 %	without waiting for a response	⊞
2,496	3.5 %	experts waiting to answer	⊞
2,395	3.4 %	are waiting to answer	⊞
2,041	2.9 %	, waiting for a response	⊞
2,038	2.9 %	not waiting for an answer	⊞
2,018	2.8 %	, waiting for an answer	⊞
70,861	100.0 %	Permalink	0.027 seconds

(b)

Figure 1: The Wordgraph visualization (a) and the textual Web interface (b)

The contributions are threefold. First, we present the Wordgraph, a dynamic graph visualization for interactive exploration of search results for complex keywords-in-context queries. Secondly, we introduce our new and scalable Netspeak retrieval engine that operates efficiently on a large corpus of text from the Web. Lastly, we perform a user study comparing Wordgraph visualization and the textual Web inter-

face, analyze query logs of the Netspeak service and investigate typical retrieval tasks related to choosing words.

2.2 RELATED WORK

In addition to the aforementioned systems, several other keyword-in-context tools exist or are being developed. Viégas and Wattenberg present the Web Seer prototype [118], which allows one to contrast the query suggestions of the Google Web search engine for two different queries. The visualization encompasses two trees whose roots represent one of the queries each, while the children represent the suggestions obtained from Google. Shared suggestions are unified, thus visualizing tree similarity, while edge thickness and node positions tell something about how often a suggested query has been posed. Paley's Textarc [76] visualizes the sentences of a text centrifugal along an ellipse shape. Frequent words of the text are depicted inside the ellipse. The legibility of individual phrases is limited with this approach.

C. Harrison [41] has generated static word graphs from small portions of the Google *n*-gram corpus as showcase examples. However, no means is provided to generate these visualizations on demand, and it also lacks interaction so that it can only be viewed as is. Collins *et al.* [16] visualize the text produced by automatic machine translation tools in the form of lattice graphs in order to support translators. Uncertainties of the tools in choosing the right translation for a word are represented by alternative paths in the lattice graph, where the commonness of an alternative, as determined by the language model underlying the machine translator, is encoded by size and shade of the nodes and their edges. Here, however, no manual wildcard queries are possible. Although our system displays a graph instead of a tree, the Degree-of-Interest Trees (DOITrees) by Heer and Card [43] and the SpaceTree by Plaisant *et al.* [81] provide some convenient patterns for the navigation of different levels of detail, supported by animated transitions in huge tree structures.

Corpora of *n*-grams are frequently used in natural language processing and information retrieval in order to support computational linguistics [64], such as data-driven error correction [60] or query segmentation [38]. While *n*-grams are usually exploited in a preprocessing fashion or to fully automate an analysis task, with Netspeak we have been among the first to consider literal *n*-grams by offering them directly as search results to the user. To the best of our knowledge, only the recently published Google Books *n*-Gram Viewer goes into a similar direction [67]. This viewer targets researchers in the humanities who study language use over time. However, it does not offer wildcard search capabilities, and its interface requires expert knowledge.

Linguistic search engines that provide query operators comparable to Netspeak include WEBASCORPUS [29] ,WEBCORP [88] as well as PHRASESINENGLISH [28],

and LSE [89]. Cafarella *et al.* [9] implement a search engine that allows to formulate parts-of-speech queries. The aforementioned approaches target linguistics researchers, for whom scalability as well as performance of the implemented indexes is only of secondary concern. Other related work can be found in the field of string processing where researchers study the scale-up of regular expression search for large text databases [15]. This body of work, however, aims at full text search, allowing for complete regular expressions, which brings about various problems in terms of runtime complexity and space efficiency. Again, the user of a regular expression search engine is different from ours. Since Netspeak and its visualization target the casual and average writer, we put strong emphasis on performance while focusing on a reasonable palette of search options to formulate queries against a database of short phrases. These constraints are exploited within our retrieval algorithms, and we are the first to study efficient wildcard search on n-gram databases.

2.3 NETSPEAK

The Netspeak text interface provides a straightforward way to search for phrases [128]. It is designed to conform to current and traditional best practices of Web interfaces for search engines, with an emphasis on simplicity and minimalism (Figure 1, bottom). It utilizes a query language that is defined by the grammar shown in Table 1.

A query is a sequence of literal words and wildcard operators, wherein the literal words must occur in the expression sought after, while the wildcard operators allow specification of uncertainties. Currently five operators are supported: the question mark, which matches exactly one word; the asterisk, which matches any sequence of words; the tilde sign in front of a word, which matches any of the word's synonyms; the multiset operator, which matches any ordering of the enumerated words; and the optionset operator, which matches any one word from a list of options. The interface displays the search results for the given query as a ranked list of phrases, ordered by decreasing occurrence of absolute and relative frequencies. This way, the user can be more confident when choosing a particular phrase by judging both its absolute and relative frequencies. For example, a phrase may have a low relative frequency but a high absolute frequency, or vice versa, which in both cases indicates that the phrase is not the worst of all choices. Furthermore, the textual Web interface offers example sentences for each phrase, which are retrieved on demand when clicking on the plus sign next to a phrase. This allows users who are still in doubt to get an idea of the larger context of a phrase.

The Netspeak web service has been publicly available with a textual interface since 2008. The analysis of 50.000 queries of Netspeak's log files reveals that the average query length (words or wildcards) was 3.3 tokens (see Table 2). The most used wildcards are the asterisk and the question mark (over 90%), the synonym-operator is used in

query	=	$\{$ word | wildcard $\}_1^5$
word	=	apostrophe letter $\{$ alpha $\}$ | " , "
letter	=	" a " | ... | " z " | " A " | ... | " Z "
alpha	=	letter | " 0 " | ... | " 9 "
apostrophe	=	" ' "
wildcard	=	" ? " | " * " | synonyms | multiset | optionset
synonyms	=	" ~ " word
multiset	=	" { " word $\{$ word $\}$ " } "
optionset	=	" [" word $\{$ word $\}$ "] "

Table 1: EBNF grammar of the Netspeak query language.

less than 5% of the queries and optionset as well as multiset are hardly used. Interestingly, the fraction of queries that do not contain any wildcards is about 20%, so that in turn, an almost 80% of the queries do. Queries without wildcards supposedly only check for the existence or the commonness of a phrase. More than 70% of the query phrases include only one wildcard (see Table 3).

An in-depth analysis indicates interesting patterns of user behavior. Most users interact with Netspeak in sessions (i.e. by posing a series of queries within a certain time frame). Only 18% of the queries belong to single-query sessions. We have identified two different session types:

(1) BUNCH OF QUERIES. In this case, a session consists of unrelated queries, where none of the queries have words or wildcards in common with previous or successive queries. It appears as if users first write a large chunk of text and then check those phrases about which they are uncertain.

(2) QUERY REFINEMENT SESSION. In this case, a session consists of related queries, where queries following each other are very likely to have words and wildcards in common. It appears as if a user is working on a particular phrase, searching for alternatives. This session type is most common.

The average number of queries per session is 5.6 and the average duration of a session is about 6.5 minutes. A few sessions took very long indeed, lasting more than half an hour. In some of the refinement sessions the users obviously struggled with a certain phrase and continually exchanged words and wildcards over a long period of time. Long refinement sessions are sometimes interrupted by unrelated queries and then continued later on.

Netspeak's textual interface does not support the concept of sessions. Thus query refinement sessions require the user to memorize the results of already performed queries and relate them to each other and to further queries in his mind. This is a

Query Length	1	2	3	4	5	≥6
Fraction	6.2 %	13.9 %	53.5 %	15.9 %	8.08 %	1.9%

Table 2: Relative fractions according to query length (from Netspeak's query log).

Wildcards	0	1	2	3	≥4
Fraction	20.7 %	71.5 %	7.09 %	0.60 %	0.11 %

Table 3: Relative fractions according to the number of contained wildcards (from Netspeak's query log).

challenging cognitive task and was a strong motivation for the development of the Wordgraph. The Wordgraph interface allows the user to start with a simple query, which can be visually refined and extended while the word graph visualization shows animated transitions between the changing result sets.

2.4 WORDGRAPH

The Wordgraph visualizes the resulting *n*-grams of a query in a layered graph (Figure 1, top) and offers interactions with the result set. The nodes of the graph correspond to the words of the *n*-grams, and an edge represents the connection between two subsequent words of an *n*-gram. Consequently, each *n*-gram of a result set is represented as a path through the graph. The layers of the graph are arranged in vertical columns to facilitate reading. Every column corresponds to one element of the query, which can be a literal word or a wildcard character, as defined in Table 1. Multiple occurrences of the same word in a column are merged into a single node.

The graph can be drawn in two ways, a *split path view* and a *condensed path view* (Figure 2). The *split path view* displays the *n*-gram paths of a result set independently—similar to the text view—and reveals the overall complexity of the result set. The *condensed path view* merges shared subpaths of different *n*-grams, which is a compact abstraction of the result set, where individual *n*-grams are no longer directly visible. While merging the paths significantly enhances overall legibility of large graphs, it also brings about the problem of spurious results since more paths can be created than are actually supported by the result set. However, the condensed path view is the preferred view in practice, which is why we have developed several techniques to counter this problem and to allow users to interactively explore the result set.

While the text interface of Netspeak offers no interaction beyond the retrieval of example sentences for a particular *n*-gram, the Wordgraph provides various means

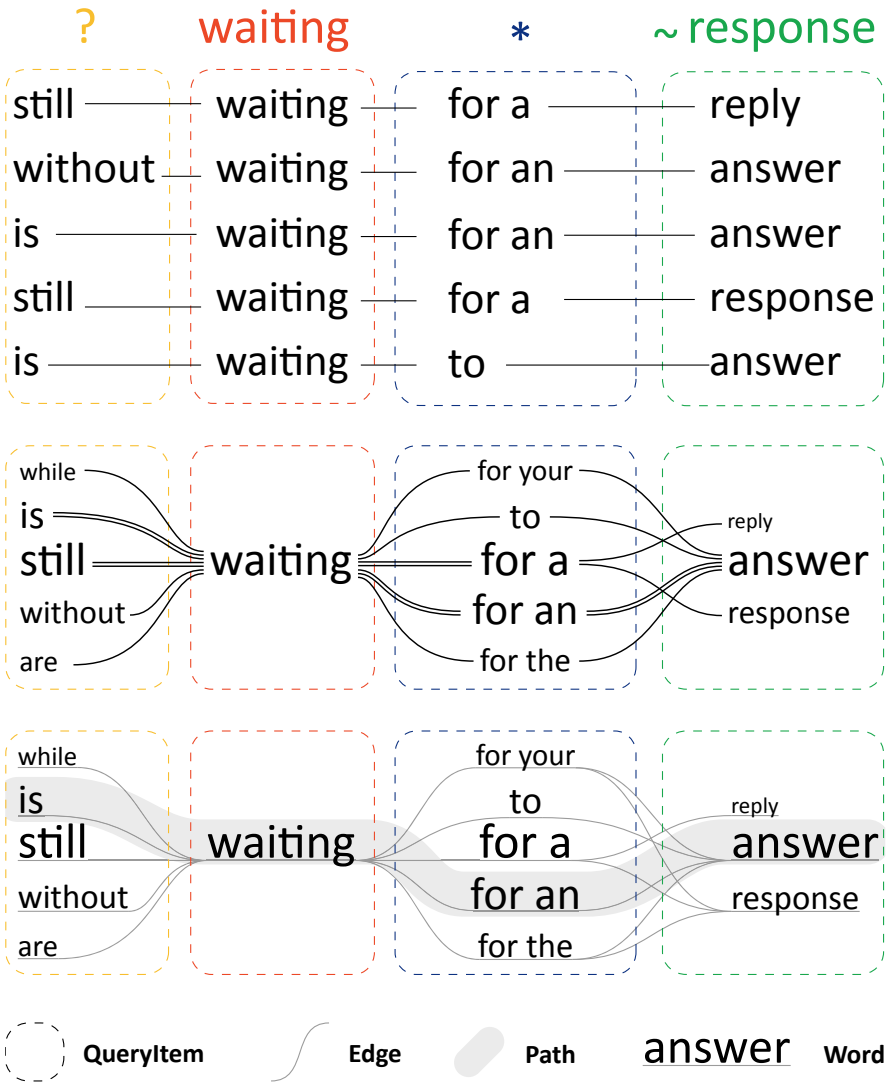


Figure 2: The query ? waiting * ~response combines one word and three wildcards; the search results are shown. The Wordgraph visualizes *n*-grams as paths through the graph, merging multiple occurrences of a word within a column. Every path can be drawn individually (split path view, middle) or shared subpaths can be merged (condensed path view, bottom). The latter view increases the probability for spurious results. That is, a path for `is` waiting for a reply is shown although this 5-gram is not part of the result set.

for exploring the search results, including filter techniques and support for navigation. Even more importantly, visual query expansion and query modification are supported along with animated transitions among subsequent result sets.

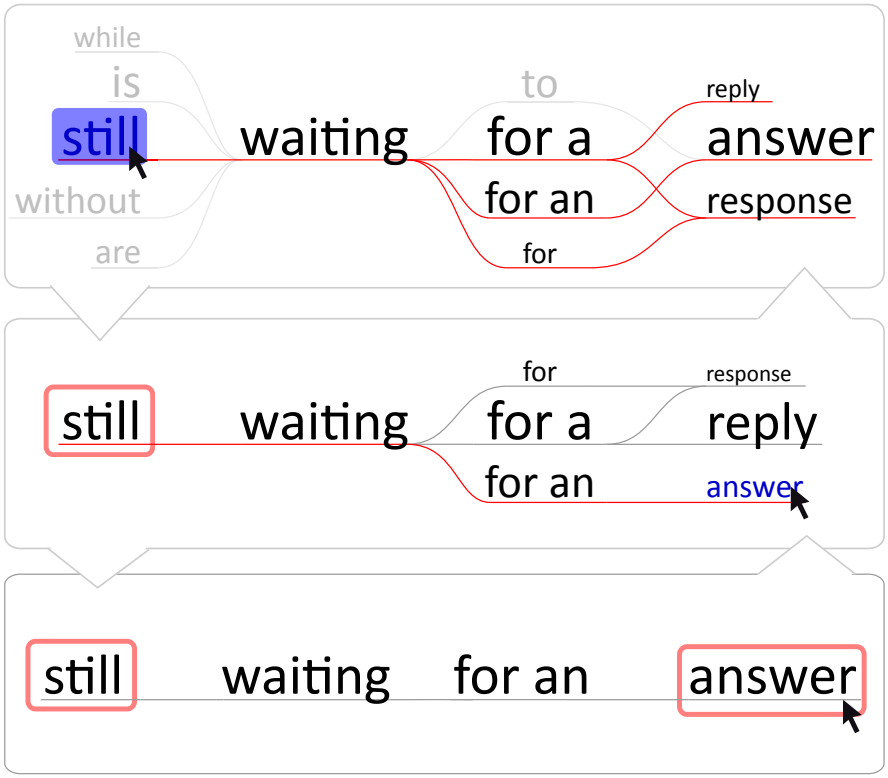


Figure 3: The Wordgraph offers several filter operations: (1) Hovering the mouse above a node highlights all n-gram paths passing through the node. (2) Selecting a node deemphasizes all paths of n-grams that do not contain the selected word. Multi-selection is supported. (3) The subgraph filter hides elements that do not belong to selected paths.

2.4.1 *Graph Filter*

The filter operations allow users to reveal the paths passing through a certain node, to emphasize certain paths, and to select a subgraph by specifying a set of nodes (Figure 3). The filter operations are orthogonal, as in they can be applied repeatedly in an arbitrary order. Users may also switch between the condensed path view and the split path view at any time. Transitions between different views are animated to facilitate the understanding of the relationships between the different layouts.

2.4.2 *Horizontal Query Expansion*

Since the basis of our retrieval engine is the Google n-gram corpus, only n-grams up to a length of n 5 words can be retrieved. By means of our query expansion technique we can address this limitation and allow the retrieval of longer phrases based on

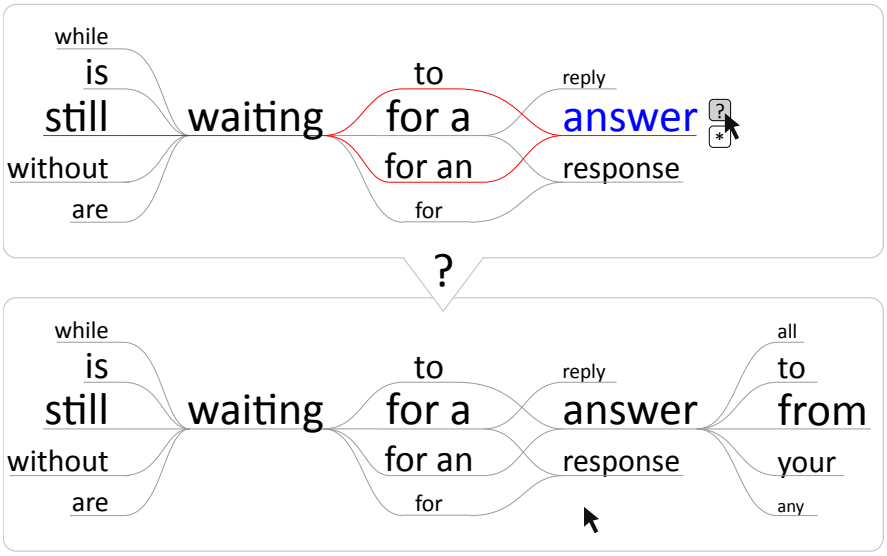


Figure 4: Horizontal query expansion: by clicking on the wildcard icon next to the word answer, a set of queries is generated for all n-grams whose paths pass through this word's node, complemented by the respective wildcard. The n-grams retrieved with these queries are integrated into Wordgraph which results in a new column.

those already displayed in Wordgraph (Figure 4). By clicking on the expansion icons shown next to a word while hovering over it, new queries are constructed for all n-grams whose paths go through the word's node, using up to four preceding words and appending the respective wildcard ? or $*$. The union of the result sets of all these queries is then integrated into the existing graph structure, and new columns are added as needed. Every expansion entails Ok^3 new queries, where k denotes an upper bound on the number of incoming edges of a word in the Wordgraph with k typically being between 4 and 10. The n-grams formed in this way may be longer than the n-grams contained in the underlying corpus and thus may be incorrect or meaningless. Nevertheless, sensible results have been observed in many cases.

2.4.3 *Vertical Query Expansion*

In addition to the horizontal query expansion, we provide a means to vertically expand the graph (i.e. within a column) by replacing words by wildcards or wildcards by other wildcards. We distinguish between operations acting on a single word and operations that are related to an entire column (Figure 5). Currently, we offer only the synonym-operator for word-related query expansion and the wildcards ? and $*$ for column-related query expansion, which replace the word or operator in the corresponding column of the original query. In principle, for both cases all three operations could be made available. For word-related expansions the word is replaced by the corresponding wildcard in all paths through this word. Column-related query ex-

pansion could be transformed simply into word-related query expansions by applying the chosen wildcard to each word in the column.

For each expansion the corresponding queries are generated and the results are integrated into the graph structure. The word-related operations just add the new *n*-grams to the existing structure. Column-related expansions transform the graph into a new one by removing paths only contained in the old result set, preserving paths that exist in both sets, and adding new ones.

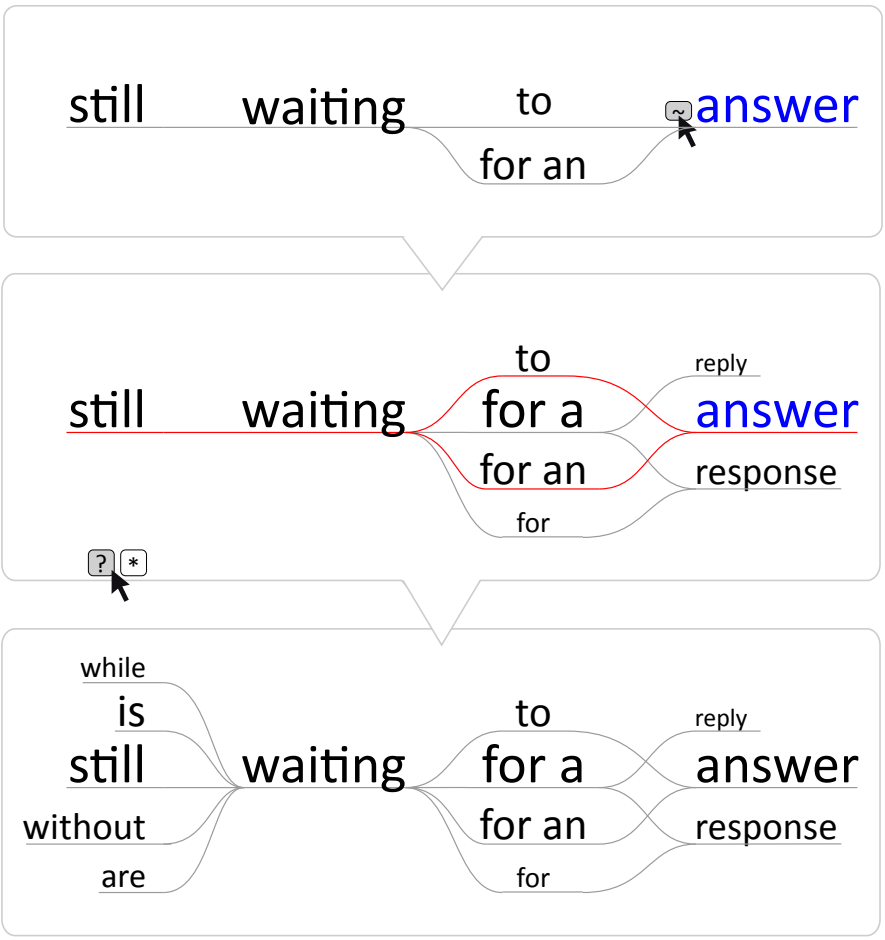


Figure 5: Vertical query expansion. The user starts with a simple query containing one wildcard (top). In the next step the query is expanded with synonyms of the word answer and the query results are being integrated into Wordgraph (middle). The third step changes a column representing a query word into a wildcard column (bottom).

Horizontal and vertical query expansion lead towards visual query specification and modification, which replaces the common sequence of manually typed-in queries. The animated transition between the query results visually relates the results of the subsequent queries to each other instead of simply replacing the previous result set by a new one.

2.4.4 *Navigation*

The query expansion technique produces word graphs that have too many columns to fit on the screen. To allow for navigation, we implemented horizontal panning and scrolling support by directly dragging the entire graph. An overview bar at the bottom of the screen (Figure 6) helps the user to control the horizontal panning and make it possible to jump immediately to a specific column, which automatically scrolls into the center of the screen. Columns which do not fit on the screen appear collapsed on the overview bar.

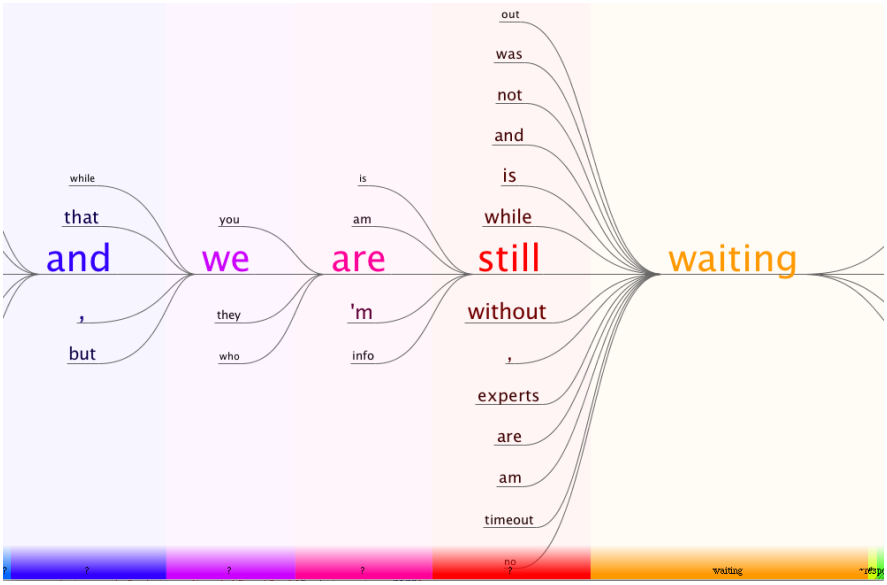


Figure 6: Horizontal navigation: The overview bar at the bottom of the screen shows the columns of Wordgraph. Selecting a column moves that column into the center using an animated transition. Also, the whole graph can be moved horizontally while columns are collapsed and expanded as necessary.

Vertical navigation becomes necessary if a column does not provide sufficient space for the set of retrieved words. We experimented with two different strategies for dealing with this case: an explicit focus-and-context approach and a clipping technique (Figure 7).

Our focus-and-context technique distorts the font size in the distal areas of the column, but leaves it unchanged in the central 80% of the column. This is an important design decision, since we map the relative frequency of a word to the font size. With the simple clipping technique the clipped edges hint at further words located outside of the visible area just as the focus-and-context technique does. However, it completely clips edges that connect clipped words in two subsequent columns. Both techniques generate overplotting of the edges connecting to the distal words. The overplotting is more pronounced with the focus-and-context technique, which makes

it more difficult to estimate the number of words in the context area (Figure 7). As a result, we generally prefer the clipping technique over the focus-and-context technique.

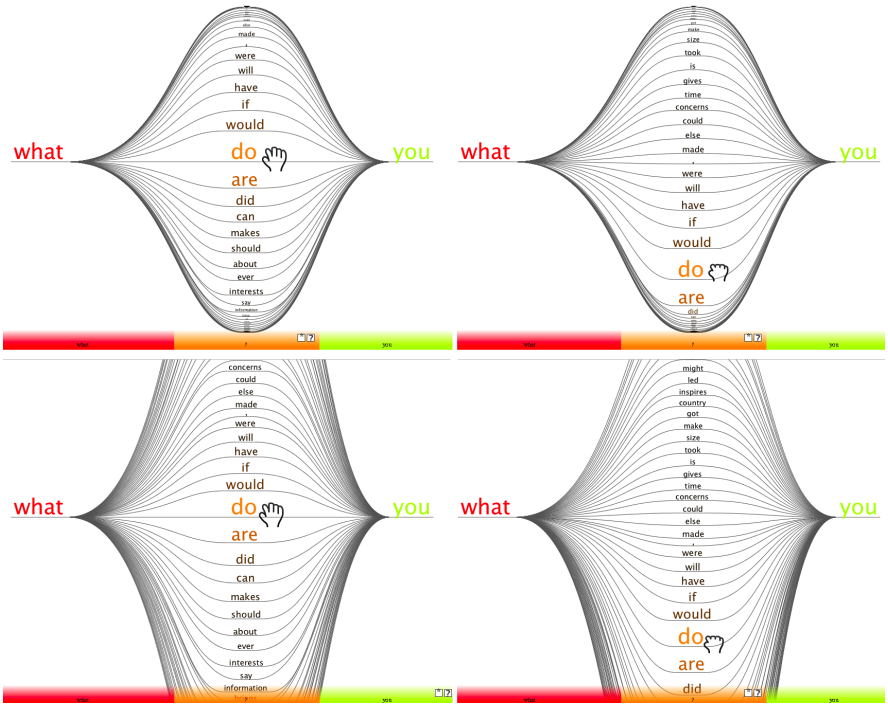


Figure 7: Vertical navigation: (top) The focus-and-context approach has an aesthetic appearance. (bottom) The simple clipping solution results in a clean separation of the edges connecting to the distal words. The left images show the initial view, while the right images show the resulting view after dragging the word **do** downwards for shifting the focus on the words in the upper part of the column.

2.5 WORDGRAPH LAYOUT DETAILS

This section explains the important design decisions for the layout and rendering in Wordgraph. The central concern is the legibility of phrases, and hence the placement of words in subsequent columns is essential. The layout also needs to reflect properties of individual words (e.g. font, size, color and opacity, see Figure 8), properties of edges (e.g. path, color and width) and attributes of *n*-grams (e.g. absolute and relative occurrence frequency).

The layout process consists of five steps:

(1) Horizontal partitioning of available screen space into columns.

(2) Vertical ordering within these column;

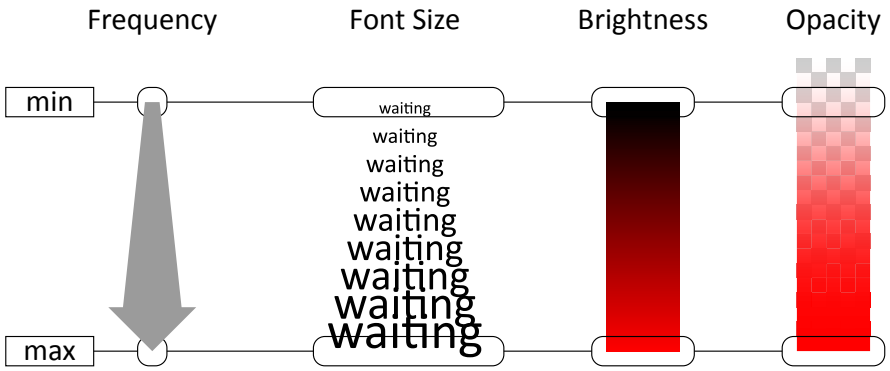


Figure 8: The accumulated occurrence frequency of the individual words are mapped by utilizing several features to visualize the importance of a single word among all words: (1) the font size, (2) the brightness of the font color and (3) the opacity of the font color (not default, because it seems to depend on individual preference).

(3) Exact placement of words.

(4) Drawing of edges between (underscoring) words.

(5) Performing crossing reduction, if possible

2.5.1 *Screen Partitioning and Word Placement*

The initial layout of Wordgraph evolves from the submitted query. The longest *n*-gram returned determines the number of necessary columns. The width of each column takes into account font sizes, word lengths and additional padding, as shown in Figure 10. Within a column, each word is horizontally centered, except for the first and last column respectively.

The vertical arrangement can be done in two ways: one strategy is the *top spread ordering* (Figure 9, top), which is similar to the text view. The second strategy is the *center spread ordering*, which places words in a column with decreasing font size, starting from the center and alternating the placement above and below (Figure 9, bottom). The latter strategy is preferred since it places the most important query result in the middle of the screen and facilitates the tracing of alternative phrases without introducing large inter-column skips.

For the vertical word placement within a column we experimented with two possible layouts, shown in Figure 10: the *maximal word spreading* uses the entire vertical and horizontal space of a column for equally distributing the words; it is independently applied for each column. The alternative *grid-based word placement* is more compact and uses a grid to place the words. The defined cell height for all columns depends on the font size of the most frequent word of all columns. In every column the algorithm starts from the center and places the words above and below, aligned to the defined

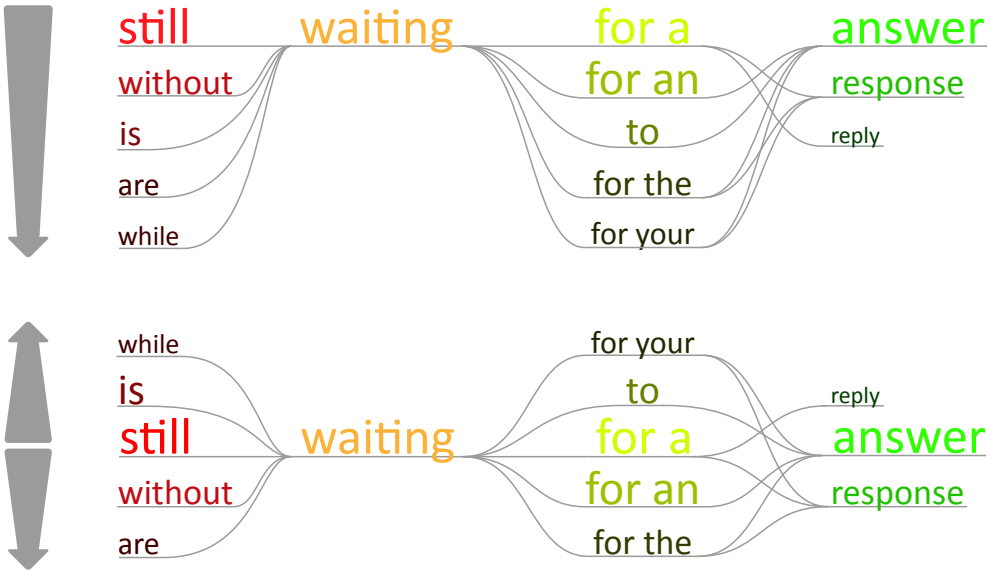


Figure 9: Arranging words in a column according to their frequency: The *top spread ordering* (top) and the *center spread ordering* (bottom). The relative occurrence frequency of a word in a column is mapped to its font size, color and brightness.

cell height (Figure 10, bottom), which minimizes the vertical spread from the center. Horizontally the algorithm is more flexible: in the first and the last column the words are aligned to the inner padding, while in the other columns they are centered. We found that the grid-based vertical partitioning of all columns along with a minimal spread from the center (Figure 10, bottom) facilitates the readability of the phrase fragments since it resembles a printed page.

2.5.2 *Edge Drawing*

A path represents an n-gram within the graph structure. Therefore, it is also a sequence of words connected by edges. A word only occurs once per column, so many paths could be incident to a node.

As previously mentioned in section 2.4, the edges of Wordgraph can be rendered in two different ways (Figure 2). The condensed path view draws a direct representation of the graph with at most only a single edge between words. The split path view shows all n-grams contained in Wordgraph by drawing all the edges of the n-grams into the graph. Each edge is defined by a cubic Bézier curve. The start point and end point are located at defined locations (ports) on the source word and the target word. The tangents at these points are always horizontal to allow for a smooth transition from a straight line through the word into the edge.

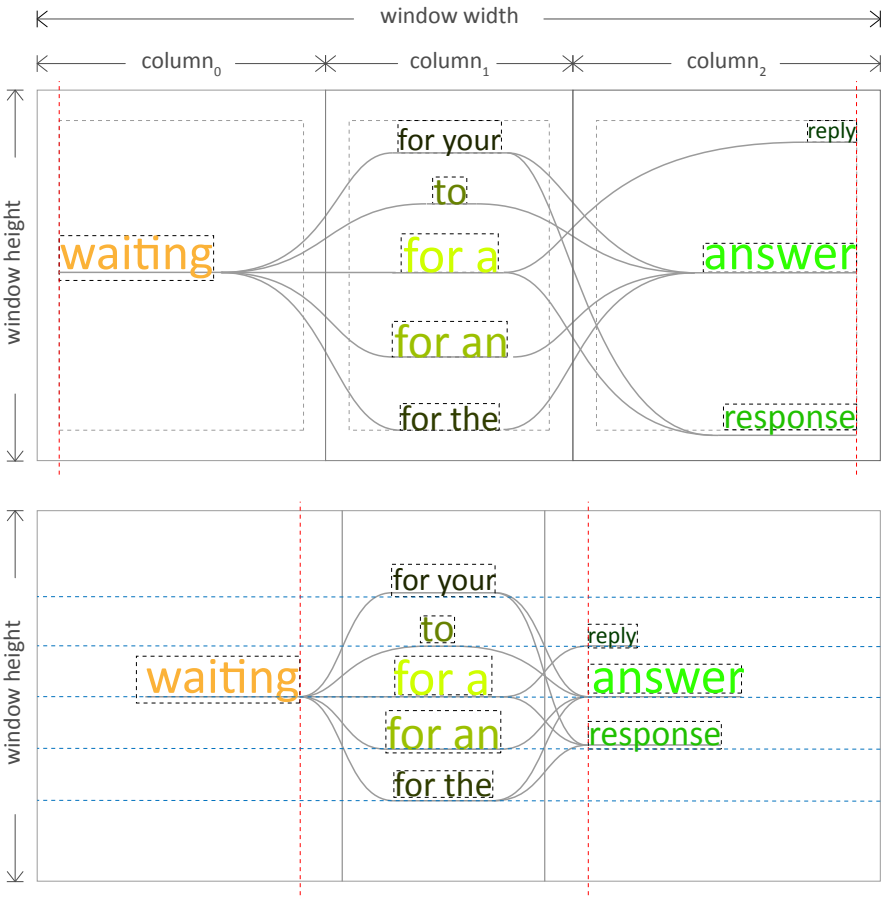


Figure 10: Column layout and word placement in a column. Maximal word spreading (top). Grid-based word placement of all columns (bottom).

The *condensed path view* places the port for connecting edges at either end of the baseline of the word. The baseline of the word itself is drawn such that the line passes below the word to the other port and continues to an outgoing edge (Figure 11.1). We found that drawing a continuous line below the words, which connects incoming and outgoing edges, significantly contributes to the readability of phrase fragments. Moreover, interrupting the curves by words is recognized as a set of single words without meaning rather than a coherent phrase.

The *split path view* shows all paths defined by the *n*-grams from the search result set at once. The paths are also drawn in the background of the words such that tracing of an individual path across multiple columns is fully supported. Figure 11.2 and 11.3 show two different ways of vertically arranging the incoming and outgoing edges of a word. Figure 11.2 attaches the incoming and outgoing edges of a path to ports at the same vertical position and avoids crossings behind the word, but introduces additional crossing outside the word. Alternatively, incoming and outgoing edges on both sides are attached to appropriate ports depending on their starting position (Figure 11.3).

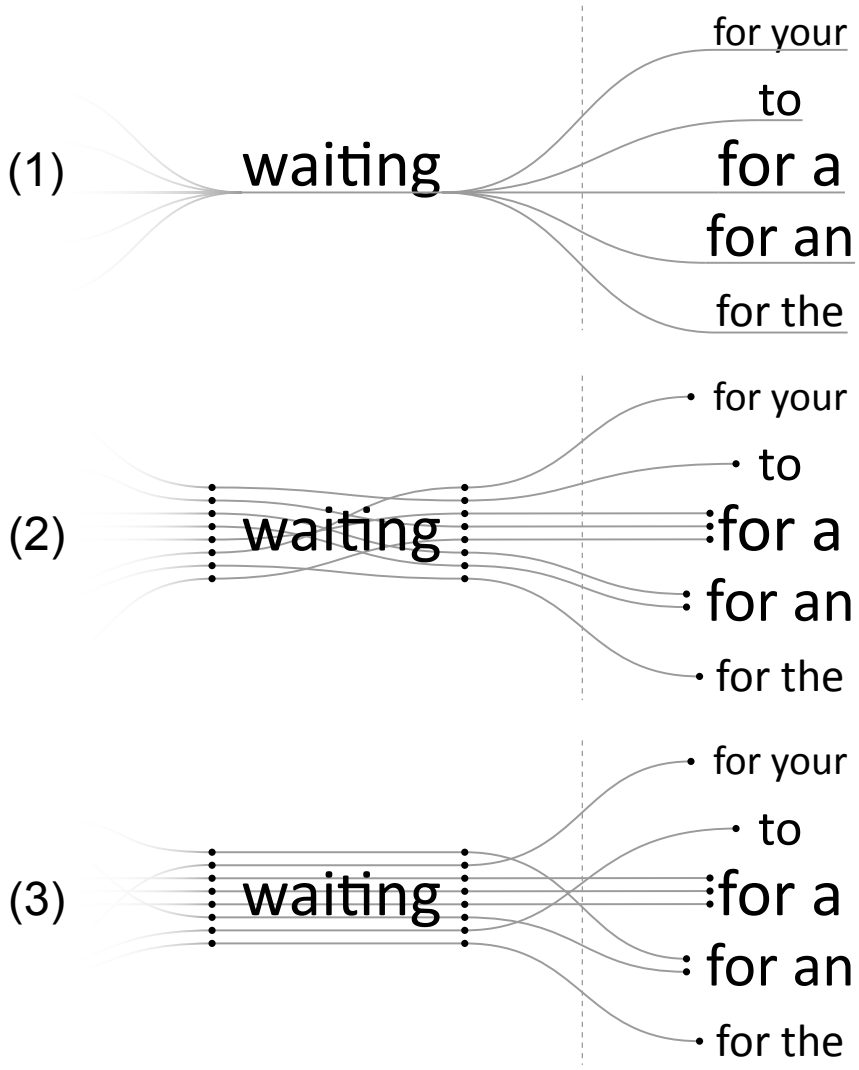


Figure 11: Possibilities for edge drawing. The edge ports are marked as small dots. (1) *Condensed path view:* all paths between two words are drawn as a single edge and the incoming and outgoing edges are connected by a line passing below the word to improve readability. (2) *Split path view:* each path is drawn independently. Crossings occur in the background of the words. (3) *Split path view:* each path is drawn independently. Crossings occur after the words.

In this case edge crossings occur behind the word, which was generally preferred particularly in combination with the available interaction techniques.

2.5.3 *Edge Crossing Reduction*

Edge crossings between columns are introduced when merging multiple occurrences of a word in a column. This is particularly annoying if a node in the upper half of a column is connected to a node in the lower half of the subsequent column. For the *center spread ordering* there is some potential to minimize the number of edge crossings. Our approach is inspired by the classical algorithm for drawing layered graphs which was suggested by Sugiyama [107] for his barycenter-based layer-by-layer sweep.

The Wordgraph itself consists of columns which form a horizontally oriented layered graph. Thus, to reduce the number of crossings, each possible pair of layers can be treated by fixing one layer and permuting the other employing a heuristic (often barycentric or median) which reorders the nodes according to the positions of their counterparts in the fixed layer. However, in our case, the order of descending node size away from the center should not be lost, and therefore the nodes cannot be reordered arbitrarily.

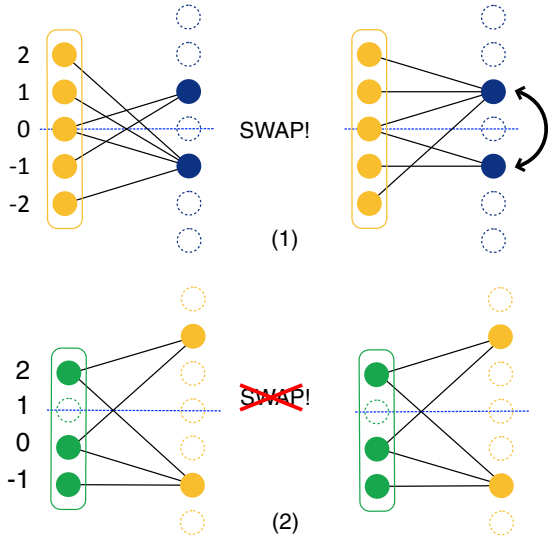


Figure 12: Edge crossing reduction between two columns. Swapping two equidistant nodes might reduce the number of crossings (1) or it does not (2).

Edge crossing reduction algorithm. Let $G \ \langle V,E\rangle$ denote a Wordgraph. We use a layer-by-layer sweep approach (from left to right) to process the columns of G with respect to their predecessor. Given the i-th column $V_i \subset V$, with $V_i \ v_1,\ldots,v_l$, its preceding or succeeding column V_j, respectively, as well as the edges $E_{ij} \subset E$ between them. Presuming the nodes in V_i have already been ordered according to the center spread layout, say, $v_c \in V_i$ is center node, our crossing reduction algorithm assigns ranks to the nodes in V_i. A mapping $rank_i \ V_i \to \{-\lfloor l/2\rfloor,\ldots,\lfloor l/2\rfloor\}$ is set up to map the nodes in V_i onto ranks, where a node's rank denotes its distance to the cen-

ter node v_c and the sign of a node's rank denotes whether it is above or below v_c. Likewise, $rank_j$ assigns ranks to the nodes in V_j (Figure 12). Then, for each pair of equidistant nodes $v, v' \in V_i \times V_i$, where $rank_i v \; -rank_i v'$, it is determined whether they should be swapped within V_i, which is the case if the barycenter of v' lies above that of v:

$$\sum_{\{v',u\} \in E_{ij}} rank_j u \quad \geq \quad \sum_{\{v,u\} \in E_{ij}} rank_j u.$$

We tested different orders to process the Wordgraph layers using several graphs from queries containing different numbers, kinds, and positions of wildcards. Altogether, we found that the simplest approach to process the layers from left to right yields the best results on average (mean of 26% crossing reduction). For simple graphs our algorithm performs only a few swaps. However, for complex graphs a reduction of crossings of up to 52% was observed.

2.5.4 *Layout Guidelines*

Based on our experience with alternative implementations of the Wordgraph interface we derived the following list of layout guidelines. These guidelines might also be useful for other word-based visualization approaches.

- *Center spread ordering* works better than *top spread ordering*.
- Implement a vertical grid to align words across different columns.
- A minimal vertical word placement starting from the center is preferable.
- Underlining emphasizes the connectivity of a collocation and improves legibility significantly.
- In the split path view, drawing edge crossings after (instead of behind) words gives a less tangled appearance.
- Crossing reduction between subsequent columns improves legibility.
- Animated transitions are essential for filtering operations, query exploration and navigation.

2.6 NETSPEAK'S RETRIEVAL ENGINE

A salient feature of the Netspeak phrase search is its efficiency at web-scale. This section introduces the underlying technology to deal with this vast—and still growing—amount of data.

Netspeak combines state-of-the-art data structures with original retrieval research in order to answer wildcard queries at the highest possible speed. At its core is a query processor that is tailored to the following task: Given a wildcard query q and a set of n-grams D, retrieve those n-grams $D_q \subseteq D$ that match the pattern defined by q. The query processor addresses the three steps indexing, retrieval, and filtering, as illustrated in Figure 13.

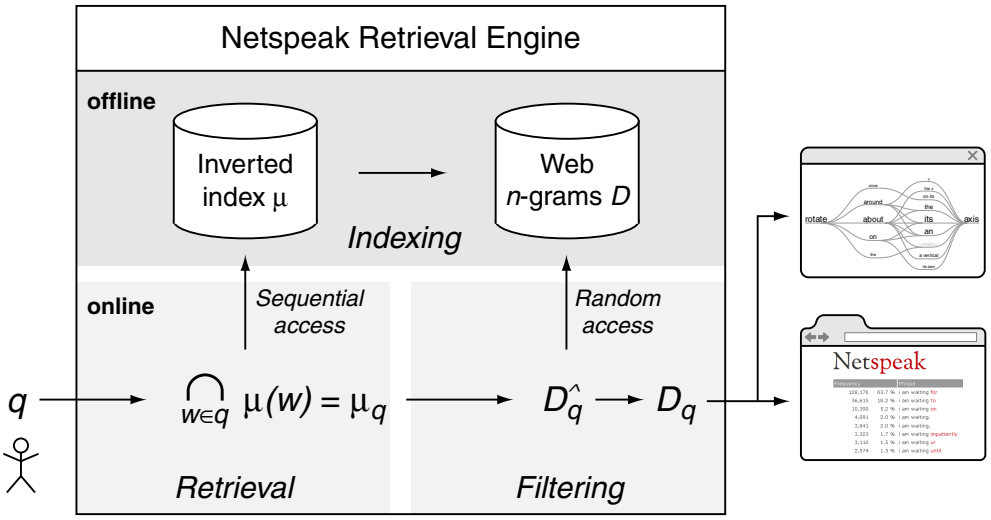


Figure 13: The Netspeak retrieval engine at a glance: Given a query q the intersection of relevant postlists yields a tentative postlist μ_q, which then is filtered and presented as a ranked list or in graph form. The index μ exploits essential characteristics that are known a priori about possible queries and the n-gram set D.

The indexing step is done offline, once before answering the first query. Let V denote the set of all words found in the n-grams D, and let $D\Theta$denote the set of integer references to the storage positions of the n-grams in D on hard disk. During indexing, an inverted index $\mu \;\; V \rightarrow \mathscr{P}D\Theta$is built that maps each word $w \in V$ to a skiplist $\mu w \subseteq D\Theta$comprised of exactly all references to the n-grams in D that contain w. μw is referred to as posting list or postlist. Since D is invariant, μ can be implemented as an external hash table with O_1 access to μw. For μ being space-optimal, a minimal perfect hash function based on the CHD algorithm is employed [4].

The two online steps, retrieval and filtering, are taken successively when answering a query q. Within the retrieval step a tentative postlist $\mu_q \;\; \bigcap_{w \in q} \mu w$ is constructed; μ_q is the complete set of references to n-grams in D that contain all words in q. The computation of μ_q is done in order of increasing postlist lengths. Within the filtering step, a pattern matcher is compiled on-the-fly from q, and D_q is formed as a set of references pointing to the matching n-grams in μ_q. Netspeak exploits the fact that the search in D described above can be significantly narrowed down in our application; the following subsections provide an overview of the developed strategies.

2.6.1 *Tailored Indexing*

Tailored indexing aims to reduce the filtering effort. The starting point is the distinction of the Netspeak queries into fixed-length queries and variable-length queries. The former contain only wildcard operators that represent an a priori known number of words, while the latter contain at least one wildcard operator that expands to a variable number of words. For example, the query `fine ? me` is a fixed-length query since only 3-grams in D match this pattern, while the query `fine * me` is a variable-length query since n-grams of length $2,\ldots,n$ match. Obviously, fixed-length queries can be answered with less filtering effort than variable-length queries: simply checking an n-gram's length suffices to discard many non-matching queries. The Netspeak query processor first reformulates a variable-length query into a set of fixed-length queries, which then are processed in parallel, merging the results. For example, the aforementioned query `fine * me` is reformulated as follows:

```
fine me
fine ? me
fine ? ? me
  ⋮
```

Since the maximum length of an n-gram in D is small ($n < 8$ for many relevant computer-linguistic applications), the number of fixed-length queries obtained from reformulating a variable-length query is tractable. With k as the number of variable-length wildcard operators in a query q, the size of the respective fixed-length query set is in On^k. In the following we assume all queries to be fixed-length queries.

A proper inverted index μ for D enables $O1$ access to n-gram sets that fulfill a certain constraint—usually a word w that must occur in all n-grams referred by μw. However, by considering also the position of w an even tighter constraint is imposed. This idea is exploited by the following index μ:

$$\mu\ V \times \underbrace{\{1,\ldots,n\}}_{n\text{-gram length}} \times \underbrace{\{1,\ldots,n\}}_{\text{word position}} \to \mathscr{P}D\Theta$$

where the preimage is the Cartesian product of D's vocabulary V, the possible n-gram lengths, and the possible positions of words within n-grams. Given the query q `fine ? me`, the postlist μ_q is defined as follows:

$$\mu_q\ \mu\text{“}\texttt{fine}\text{”},3,0 \cap \mu\text{“}\texttt{me}\text{”},3,2$$

μ_q consists of references to all 3-grams in D that have “fine” as their first word and “me” as their third word. Since this is exactly what the query is asking for, the subsequent step of filtering μ_q can be omitted.

2.6.2 *Postlist Pruning*

Postlist pruning aims to reduce set operations. Let $f\ D \to \mathbf{N}$ be a function that indicates the occurrence frequencies of the n-grams in D. Similar to μ, f is implemented as an external hash table. During the indexing step, each postlist $\mu w,\cdot,\cdot$ for some $w \in V$ is sorted in decreasing order of the occurrence frequencies of the referenced n-grams, which allows for head pruning and tail pruning.

Head pruning means to start reading a postlist at some entry within, without compromising the recall. Given a query q let τ denote an upper bound for the frequencies of the n-grams in q's result set D_q, i.e., $d \in D_q$ implies $fd \leq \tau$. Obviously, in all postlists that are involved within the construction of D_q, all entries whose n-gram frequencies are above τ can safely be skipped. We assess τ as follows:

$$\tau\ \min_{d \subseteq q} fd,$$

where d is a maximum, non-terminal n-gram in q. For example, the query

$$q\ \texttt{sounds fine ? me}$$

contains the two maximum, non-terminal n-grams "sounds fine" and "me" with the frequencies f`"sounds fine"` 45817 and f`"me"` 566617666. Since no n-gram matching q can have a frequency larger than $\tau\ 45817$, all entries of μ`"sounds"`, μ`"fine"`, and μ`"me"` whose n-grams have a higher frequency than τ can be skipped.

To efficiently determine the first entry of a postlist $\mu w,\cdot,\cdot$, $w \in q$, whose frequency drops below τ, an additional meta index μ_f is built during the indexing step, which indexes the postlists of μ. The postlist $\mu_f w,\cdot,\cdot$ comprises entries of the form fd,i, indicating that the n-gram d referred to at the i-th entry of $\mu w,\cdot,\cdot$ has frequency fd. To keep μ_f's memory footprint small, only those postlists from μ are indexed that cannot be read at once into main memory. In addition, only every i-th entry of a postlist $\mu w,\cdot,\cdot$ is indexed in its corresponding postlist $\mu_f w,\cdot,\cdot$, so that $|\mu_f w,\cdot,\cdot|\ \ |\mu w,\cdot,\cdot|/i$, where in our case $i\ 1000$.

Up to this point, the retrieval of n-grams matching a query q is exact—but, not all n-grams that match a query are of equal importance: Netspeak users look for n-grams that occur frequently on the web. Taking this fact into consideration, we apply tail pruning on postlists that are too long to be read at once into main memory. As a result, less frequent n-grams that might match a given query may be missed. Netspeak employs three tail pruning strategies: (1) stop after a specified number of matching n-grams has been found, (2) stop after a specified number of entries from a postlist has been read, (3) stop after a specified quantile of a postlist has been read. The last strategy is used to define word-class-specific pruning heuristics, since different word classes (stop words, nouns, adverbs, etc.) have a different impact on the construction of D_q. Section 2.7 reports on effects of these strategies. On demand, a pruned search can be resumed in order to retrieve the complete result set.

2.7 EVALUATION RESULTS AND DISCUSSION

In this section we provide implementation details and evaluate Netspeak's components: we report on experiments to assess the retrieval performance of our query processor, conduct a user study, and conclude with a discussion of use cases for the Wordgraph interface.

2.7.1 *Implementations Details*

The communication between Netspeak's interfaces and its retrieval engine is implemented with the Ajax paradigm, using the lightweight JavaScript Object Notation interchange format JSON. The retrieval engine is written in C/C++ and is deployed at our site, accessible through a servlet container. The textual Web interface is implemented using the Google Web Toolkit and it is deployed on the Google App Engine. The visualization client is a stand-alone application written in Java, deployed at our site. The Java scene graph project Scenario is used to manage and display graphical 2D-elements. Scenario provides convenient methods to handle different kinds of animations [95].

2.7.2 *The Web n-gram Collection*

To provide relevant suggestions, a wide cross-section of written text on the Web is required. Currently, we use the Google *n*-gram corpus "Web 1T 5-gram Version 1" [6], which consists of 42 GB of phrases up to a length of $n \leq 5$ words along with their occurrence frequency on the web in 2006. This corpus has been compiled from approximately 1 trillion words extracted from the English portion of the Web, totaling in more than 3 billion *n*-grams. Two post-processing steps were applied: case reduction and vocabulary filtering. For the latter, a white list vocabulary V was compiled and only *n*-grams whose words appear in V were retained. V consists of the words found in the Wiktionary and various other dictionaries, complemented by words from the 1-gram portion of the Google corpus whose occurrence frequency exceeds 11 000. After post-processing, the size of the corpus has been reduced by about 46%.

2.7.3 *User Study*

We performed a user study to assess the usability of our system and to learn about the user acceptance and potential improvements. In particular, we were interested in a comparison of the Wordgraph interface and the basic textual interface. Based on feedback from public demonstrations and a pilot study (described in [91]) we

derived our main hypothesis: Both interfaces perform equally well for basic keyword-in-context queries using only a single wildcard. With more than one wildcard, users generally prefer the visual Wordgraph interface.

Ten non-native English speakers with higher English education participated in our study. All of them were volunteers from an English writing course offered by the language center at the university. None of the participants were aware of the Netspeak web service or the Wordgraph visualization. During a brief introduction of the textual Netspeak interface and the Wordgraph interface, the participants could enter queries and examine the result sets with the two different interfaces.

The actual study comprised of six pairs of queries. Each pair consisted of two different queries with similar structure and the same number of wildcards. There were two pairs using one, two and three wildcards respectively. The level of complexity of the queries increased incrementally. The participants were asked by an instructor to enter one query of every pair in the textual interface and the other one in the Wordgraph interface, select the most suitable results, and to assess on a Likert scale how helpful each interfaces was for the task, from 1 (not helpful at all) to 6 (very helpful).

The order of interfaces was counter-balanced into two subgroups. One group requested the first query of a pair with the Netspeak web interface and the second one with the Wordgraph, and vice versa. Each participant repeated the procedure for each of the six pairs, which resulted overall in 120 assessed requests, 60 per interface.

During the study the instructor observed and ranked for each participant how well the Wordgraph interaction patterns were understood. The instructor also noted his impressions about the usage of the different Wordgraph interaction patterns. Afterwards, the participants were asked in a questionnaire about the general usage and their comments on limitations and desired improvements with respect to interaction and layout.

For each level of difficulty a *t*-test was conducted to compare the Wordgraph interface and the text-interface (Figure 14). We used an alpha level of .05 for all statistical tests. For one-wildcard queries both interfaces achieved similar ratings, the Wordgraph interface (M 4.75, SE .20) and for the text interface (M 4.8, SE .23), $t9$ .15, p .88. For two wildcards, however, the results indicate a significant preference for the Wordgraph interface (M 5.55, SE .21) over the text interface (M 4.25, SE .41), $t9$ 2.94, p .016. An even stronger preference for the Wordgraph was revealed for three wildcards: (M 5.5, SE .23) vs. (M 3.9, SE .28), $t9$ 3.64, p .005. These results support our hypothesis of an increasing preference for the Wordgraph interface with an increasing occurrence of wildcards in a query.

The questionnaires reveal that all participants would like to see Wordgraph being provided as an additional interface for Netspeak. Six of them even considered Wordgraph as a substitute for the textual Web interface. They assessed Wordgraph as “very intuitive” with an average of 5.1 on a scale from 1 to 6. This positive impression was

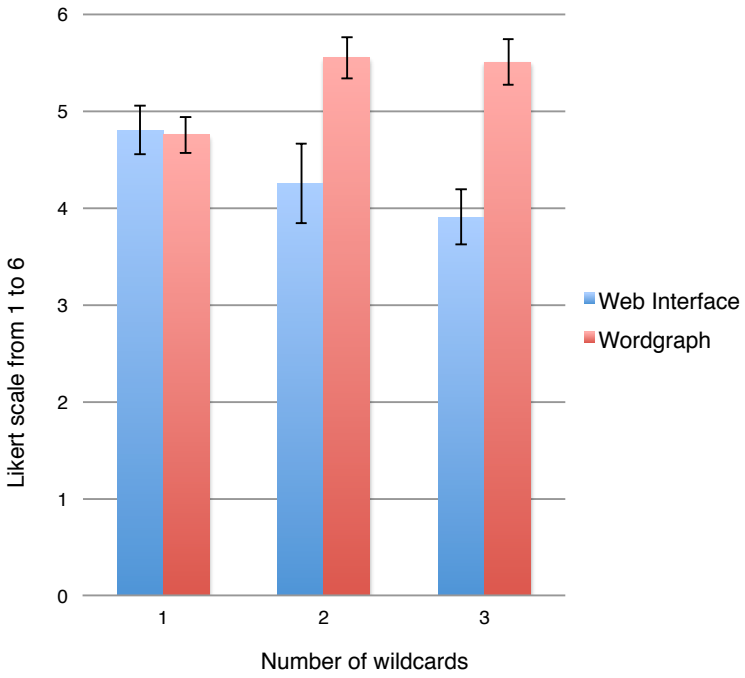


Figure 14: User Study Results: Mean interface preferences and standard errors according to different numbers of wildcards.

confirmed by the following observation: the instructors judged the understanding of the interaction patterns by the participants with an average of 5.0. The observation also revealed that seven of the participants applied the subgraph filter predominately to explore the response graph instead of using the mouse-over technique. Only one participant exclusively applied the mouse-over technique during the tests.

Altogether, the answers from the questionnaire and the general feedback we gathered during several public presentations revealed the most appreciated features of the Wordgraph: (1) Fluent result filtering. (2) Starting from an overview with the most important information. (3) The possibility of exploring the response set in detail by successive or alternating applications of subgraph filtering. (4) Finally, the improved legibility of the word sequences within the graph by following the edges through the nodes: “It gives the impression of reading from a sheet of lined paper.”

Eventually the participants suggested several interesting aspects for improving the Wordgraph interface: (1) Most users want to be able to request sentence snippets earlier during the process of exploring the graph and are not willing to wait until only one phrase remains. (2) In cases of two or more words with visually similar frequencies some users would like to see the absolute and relative frequency numbers on demand. (3) The use of thicker edges (in combination with the current coloring) for highlighting a word sequence within the graph was also suggested.

2.7.4 *Use Cases and Experiences*

Based on our query log analysis, the session types identified and the user study, we identified three practical retrieval tasks related to word choice, which have an increasing level of difficulty:

(1) *Phrase Verification.* The most basic retrieval task is to check whether a given phrase is commonly used. As mentioned above, almost 20% of all queries come without wildcards. For this task, the textual interface is fully sufficient.

(2) *Context-Sensitive Word Choice.* In this retrieval task a writer is uncertain about what alternative for a word in a given phrase is a good choice, or whether there are in fact any alternatives. This task pertains particularly to second-language speakers who often translate words using a dictionary—the exact translation of many words depends on context. In this respect, Netspeak serves as a context-sensitive thesaurus. Choosing the correct adverbs and prepositions is also a common problem. Figure 15 illustrates how Netspeak is used to find the correct collocations between the words `rotate` and `axis`.

Figure 15: Word choice with Netspeak's Web interface.

The query language of Netspeak is powerful in that it makes it possible to specify rather complex patterns of n-grams to be retrieved. A user who inserts more than one wildcard into a query is less confident about how to write a certain phrase and seeks to generalize the query in order to cover more of the possible alternatives. This, in turn, yields a longer list of results in the textual interface, which may be difficult to overview and which may not always reveal the true picture about which words to choose. Figure 15 shows an example where `about` appears in three of the n-grams, which indicates that this word should most likely follow `rotate`. The textual Web interface, however, obscures this fact and the user is forced to scan the entire result list several times to grasp the true relationships. By contrast, the Wordgraph visualization for the same

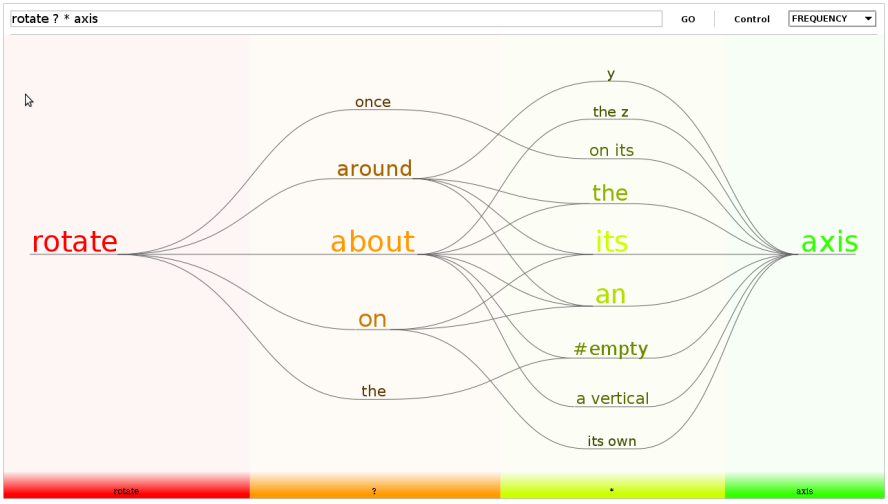


Figure 16: Word choice with the Netspeak Wordgraph.

query as above provides an overview at a glance (see Fig ure 16). This is in accordance with our user study, which revealed a preference for the Wordgraph interface over the text interface for queries containing more than one wildcard.

(3) *Exploration.* This retrieval task is about writers who want to explore the typical context of a phrase by looking at what comes before, after or in between the phrase's words. With the Netspeak's textual interface, this task is limited to exploring a context of up to four words around a query that comprises, say, only one word surrounded by asterisks. Only by means of additional queries, a user may get a broader view of a phrase's context, having to keep in mind the results of all previous queries. With the Wordgraph interface, this task is supported without further ado by means of the query expansion technique (Figure 4). The results of additional queries, which can be posed interactively, are integrated seamlessly into an existing graph so that users can construct a full picture of a phrase's context. This capability of Wordgraph is particularly useful for expert users, including linguists who investigate the characteristics of language use in a given corpus.

Remarks. While writing a text, such as a scientific paper, users often switch back and forth between different retrieval tasks. Phrase verification is the least observed task, which is documented by Netspeak's query logs; 80% of the queries comprise wildcards. There are two common types of queries: queries asking for the most suitable word in a given context, and queries asking for the typical context of a particular word or, more precisely, which common collocations a particular word has. Thus it is context sensitivity that is most relevant to the users, which is difficult to express with other commonly available tools. With the textual Web interface, one typically looks at the top results and ignores the rest—similar to the use of a Web search engine. With Wordgraph, one explores the results more thoroughly and discovers relation-

ships between words that are not apparent in the textual interface. While the latter often forces a user to formulate a sequence of similar queries, the former provides an effective means for implicit query specification, using filter techniques, query expansion and navigation.

2.8 CONCLUSIONS AND FUTURE WORK

Netspeak answers complex word sequence queries that are formulated in an expressive query language. The system is designed for efficiency and allows for real-time querying of a 42 GB text data base. The result set is explored via a textual Web interface or the graphical Wordgraph interface. Our analysis shows that the textual interface is sufficient for phrase verification and the comparison of related sentences. The Wordgraph interface allows an interactive exploration of the result set and is superior for word choice problems on complex queries. The layout of Wordgraph focuses on facilitating legibility, which is achieved by using *center spread ordering*, grid-based word placement and underscoring edges. Participants of our user study describe Wordgraph as very intuitive and appreciate the possibility of graph-based filtering during explorative analyses.

We see Netspeak in combination with its visual interface Wordgraph as a great educational tool for improving the knowledge of a second language. Additional smart operators for the query language such as antonym wildcards or semantic constraints (e.g. person names, places, dates and times) and support for further languages besides English would broaden the scope of Netspeak. An extension towards domain-specific corpora can help inexperienced authors to become familiar with the appropriate expressions and writing style in a specific field.

The interactive Wordgraph interface already allows the user to start with a simple query and visually refine and extend the query. This process generates queries containing various wildcards without the user knowing. We believe that this kind of visual query specification is the right approach since most users ($> 98\%$) of existing search engines are not aware of the simplest of search operators [129]. Further visual query refinement operations could include constraints to certain word types (e.g. parts-of-speech, location, time, colors) and recently added operators of the Netspeak retrieval engine.

Individual documents or even entire corpora can be represented as a wordgraph. An individual node in such a large wordgraph could represent a single word, a common collocation or an n-gram of a certain length. The different levels of granularity allow a trade-off between the number of nodes and the number of edges in the wordgraph, which a multi-layered graph could tie together in a single visualization. These examples illustrate only a fraction of the untapped potential, which is why we believe that the wordgraph is one of the most promising tools for semantic text analytics.

Part 3

THE PRODUCT EXPLORER: MAKING PURCHASE DECISIONS WITH EASE

The Product Explorer is an interactive parallel coordinates display for facilitating the selection process of typical products offered in online shops. Users can quickly narrow down the product search to a small subset or even a single product by using our visual query interface. Our study confirms that our interactive Product Explorer is a fast and easy-to-use tool for the product selection process of casual users.

3.1 INTRODUCTION

The major domains of parallel coordinates are still limited to the academic world. We found that occasionally companies utilize them for a specific purpose, but for the public this representation is still largely unknown. Nevertheless, the search and exploration of multi-dimensional data sets is the basis of most decisions that have to be made nowadays, such as when buying a new car or washing machine or looking for a specific building material. For these tasks, however, text-based Web interfaces for selecting products in shops are most common despite several issues: Users are not able to view all products at a glance nor estimate the overall number of products which they can choose from. In most stores just a few products are preselected. The search masks consist mainly of combo boxes or drop down lists. The search does not instantly begin while typing the search items. The resulting product list is not sortable by arbitrary attributes. Empty result sets often occur after searching a certain configuration, but most interfaces lack a possibility to inform you about the attributes that caused that situation.

These limitations motivated the development of the Product Explorer, a tool that uses an interactive parallel coordinates display to keep users well informed throughout the entire product selection process. All products and important attributes are visible at a glance and all the time. All interactions can be performed with immediate feedback and users do not have to wait until a query response appears. Our prototype is particularly suited for coping with product data by providing effective drawing of polylines and axes for easier product recognition in the parallel coordinates display. We provide a simple and intuitive visual interface and an appropriate logic for specifying attribute values and ranges for quickly narrowing down the product search to a small subset or even a single product. Product Explorer also provides effective means to cope with multiple dimensions by supporting an attribute repository and a decision bar for storing axis with already finalized configuration decisions.

3.2 RELATED WORK

Inselberg [50] invented parallel coordinates many years ago and the earliest ideas of it dating back to the 1960. A two-dimensional mapping of the multi-dimensional space is introduced by drawing each dimension as a vertical axis on a regular two-dimensional canvas. Each point in the multi-dimensional space is depicted as a sequence of line segments connecting the different vertical axes. In recent years, many important improvements with respect to interaction, drawing and data organization of parallel coordinates have been made. Siirtola [100] proposed a variety of ideas for directly manipulating parallel coordinates, for example lines can be selected and grouped with the help of logical operations. The lines within a group are visually represented by a single line, which is defined by the mean of all included lines.

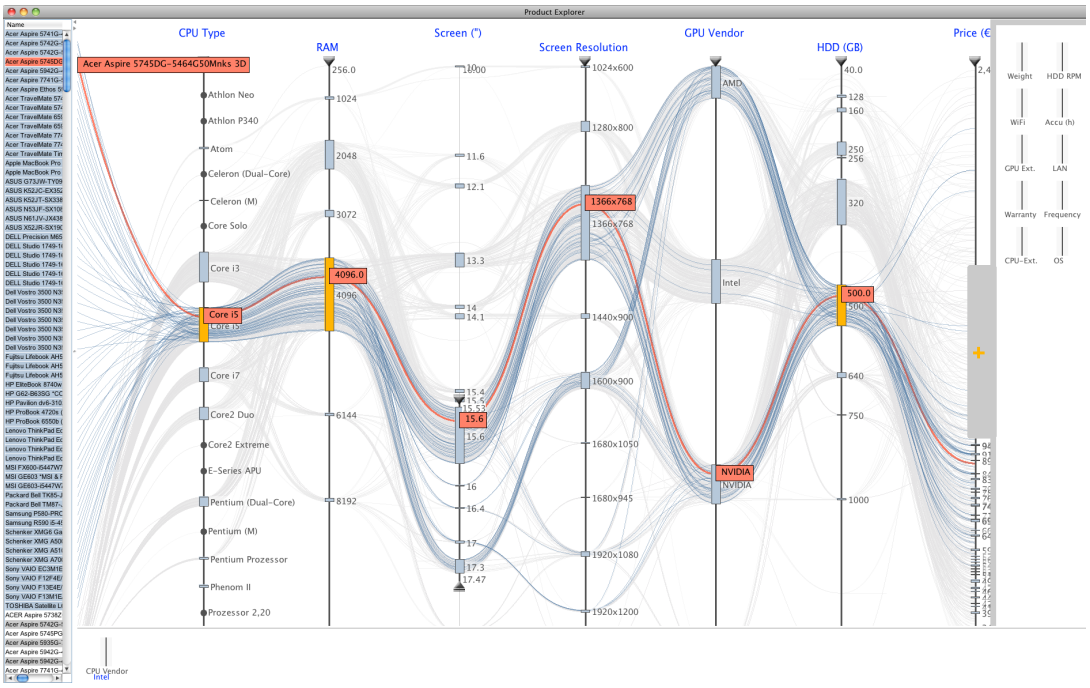


Figure 17: The Product Explorer displays a notebook data set. The user has constrained various attributes. The blue lines represent the remaining notebooks fulfilling the user's requirements. On the left a list of all notebooks is displayed showing the selected ones on top. Each list entry is linked to the parallel coordinates display. On the bottom left the decision bar contains an item, which represents an exclusive decision with respect to the CPU vendor for narrowing down the search. On the right there is an attribute repository for adding and removing axes from the display. Highlighting (in red) allows users to explore the attributes of one or more products in detail.

Graham et al. [35] proposed using a combination of quadratic and cubic curves to represent a data line. Higher order parallel coordinates [109] place invisible axes between two adjacent axes to control the curvature of the line segment. Yuan et al. [137] presented another way of controlling the curvature of a line by combining parallel coordinates, scatter plots and multidimensional scaling. Here, the points of a scatter plot are printed between two adjacent axes and used as control points for the curve. Illustrative parallel coordinates [65] go even further: McDonnell et al. focus on artistic drawing techniques to convey as much information as possible. They use various cluster techniques and adapt the edge bundles presented by Holten et al. [46]. Fanea et al. [27] combine parallel coordinates with star glyphs, by creating a three-dimensional rendering.

Heinrich and Weiskopf [44] developed a method for mapping continuous scatterplots into a density model for parallel coordinates. When using large amounts of data, it becomes necessary to structure the data. Binning can be used to gather single lines into groups. Outlier detection can further enhance the quality of the rendering [74].

Fua et al. implemented a hierarchical structure for parallel coordinates. The data is organized in a hierarchy of clusters. Advantages of structured data are an adjustable level of detail and improved performance. Dimensional reordering techniques can help to reduce over-plotting in cluttered parallel coordinate plots. Based on Peng et al. [78] dimension reorder has a significant effect on the visual expressiveness of a visualization.

Yang et al. [135] designed an interactive method for dimensional reordering, spacing and filtering based on dimension hierarchies. Reordering and spacing puts similar dimensions next to each other and reveals the structures within the data. Wong et al. developed an edge lens that allows the user to bend nearby lines in the parallel coordinate plot away from a point of interest, to reveal underlying structures in dense areas [134]. Another way of clutter reduction is random sampling of the data. Ellis et al. propose an automatic sampling lens to subsample dense areas [23]. This way trends in the data can be discovered with reduced over-plotting.

Product data attributes are partially categorical. An early approach for visualizing categorical data was introduced by Hartigan and Kleiner [30]. Mosaic plots map categories into square tiles, where the size of a tile depends on the frequency in the corresponding class. This technique was extended to three-way mosaic plots and combinations of mosaic plots and scatter plots [110]. Finally, Bendix and Kosara introduced Parallel Sets, a new technique which focuses on categorical data [59]. Relationships between different attributes are depicted by ribbons passing from one category in one axis to another category on the next axis. Categories can be arranged on each axis by the user and can be combined into new meta-categories. Elmqvist [24] offers a new presentation of scatterplot matrices by mapping the individual scatterplots on an n-dimensional cube. The user is able to navigate (like turning a dice) around this cube and visually sculpt a suitable subset of a product data base. Our approach also focuses on product data bases, but it always provides an overview of the data in one view.

3.3 VISUALIZING PRODUCT DATA

Product data bases are rarely perfect and often quite complex. In particular, if the data base is constructed by crawling the web, various attributes of individual items may be missing. A simple solution is to remove all dimensions that are not complete, but some of the removed attributes could be crucial for choosing a product. Thus, the visualization needs to be able to cope with incomplete data sets.

The attributes of a product may be continuous, categorical or ordinal. Even continuous attributes are often closer to an ordinal data type since only certain values are assumed, e.g. for the display size of a notebook. We refer to these data types as

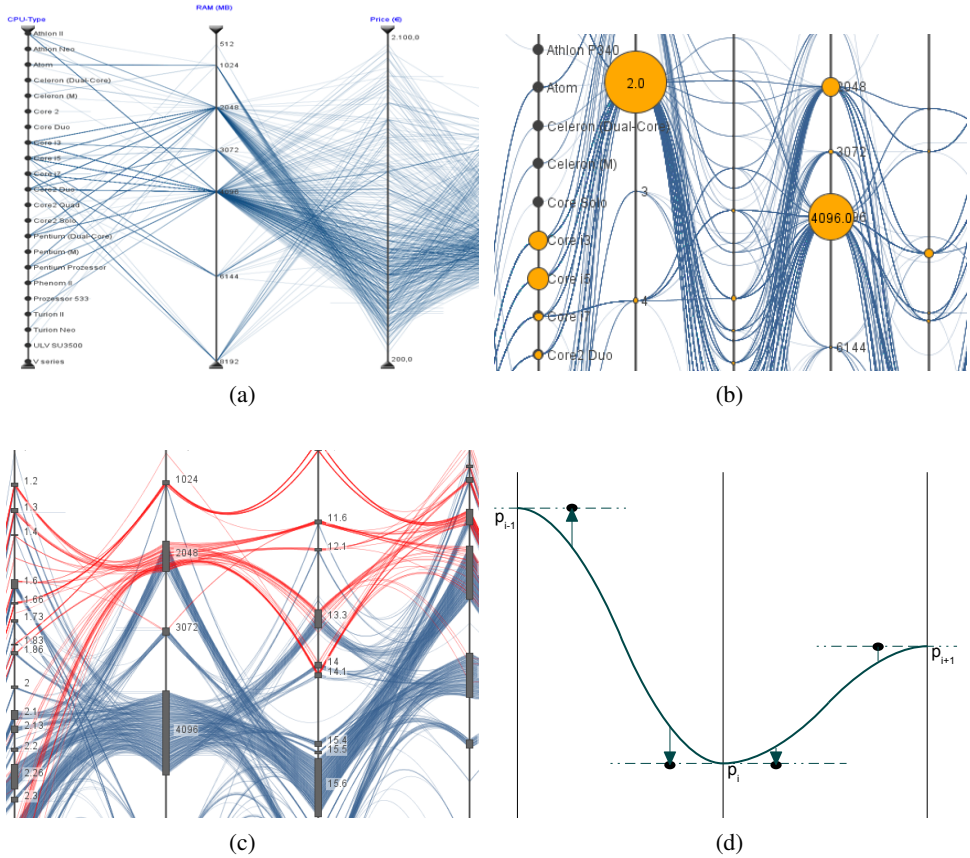


Figure 18: (a) Different kinds of axes in a classical manner of drawing: (From left to right) Categorical data, pseudo-continuous (ordinal) and continuous data axes. (b) Histograms summarize the paths sharing the same value. (c) Drawing paths as bundles. (d) Construction of a cubic curve.

being pseudo-continuous. Binary data shows whether or not a feature exists. Generally, binary data can be treated as a special kind of categorical data (see Figure 18a). In general, parallel coordinates treat each dimension as essential, but, according to product data, the attributes a product possesses are not equally important. Moreover, various attributes may be more or less important to each user for making a decision. Therefore, a global importance of an axis has to be taken into account, which is suitable for most users as well as an individual assessment needs to be enabled.

The terms clutter and over-plotting are often not very clearly distinguished, so we define clutter as a large number of axes and lines drawn together in the same system, such that it is usually overwhelming for users to recognize features and structures in the data or to interact with the display. With plain parallel coordinate drawings this issue cannot be avoided. As discussed in the related work, several techniques have been developed that try to overcome this problem.

The other major issue, particularly for product database visualization, is the handling of categorical and ordinal data. An ever-increasing amount of lines is plotted one above the other which is caused by sharing the same values in succeeding axes. Sometimes lines continue on the same path even over many axes. Thus, users are not able to recognize how many lines (i.e. items) share the same value(s) in one or more dimensions (see Figure 18a).

3.3.1 *Drawing Axes and Extended Areas*

A simple possibility to help users to clearly understand the number of products with an identical attribute is the use of round histograms, which summarize the paths sharing the same value within an axis. This technique provides a visual representation of the quantitative distribution and relations within a dimension as well as over the entire plot. Additional information can be easily displayed within, if there is enough space. However, all overlapping path segments still pass through the center point of the histogram (Figure 18b).

Thus, we prefer another approach to visualize the number of lines going through a point. The aim is to spread a particular point on an axis to a vertical area. The vertical height of a point is determined relative to the number of lines that pass through the point. This configuration also shows how many products have a certain value and reduces the over-plotting situation at crowded points by distributing the start and end coordinate of line segments between the axes to a vertical range (see Figure 1). The fraction of an axis that can be used for extended areas is globally adjustable.

However, we believe that there is an optimal range for this fraction. Either the areas are too small, so users are still not able to recognize the amount of lines going through or the areas are too large such that the users lose the mental association of an area as a point. Graham [35] proposed a similar possibility to vertically extend points. Contrary to our aim, which is the visualization of frequencies along an axis, his solution is motivated by aesthetic reasons to improve the curvature of a line across the axes. Also the interaction is used to adjust the curves by spreading the already extended points further. Instead, we use the extended areas for an explicit selection of an attribute value.

3.3.2 *Visualizing Missing Data*

As previously mentioned, product data bases are rarely complete with respect to the set of potential attributes. Therefore, we have to visually represent gaps within the data set. Instead of introducing a pseudo-category *undefined* on each axis and avoid giving the false impression that there exists a value on an axis, we do not draw a line or a curve across the axes which do not contain a value for the current line.

We propose two simple possibilities to circumvent this problem (Figure 19). The first one just indicates a line by drawing a stub, which does not cross the axis where the value is missing. In combination with extended areas, the occurrence and the amount of missing values can be easily estimated, but which stubs correspond with which counterpart on the next axis can only be assumed (Figure 19 left).

Thus, we propose a second possibility in which a translucent line with decreasing opacity links the two stubs (Figure 19 right). The barely-visible line spans between the inconsecutive axes and connects at the appropriate values, but it disappears behind the axis that does not contain a value for the given line.

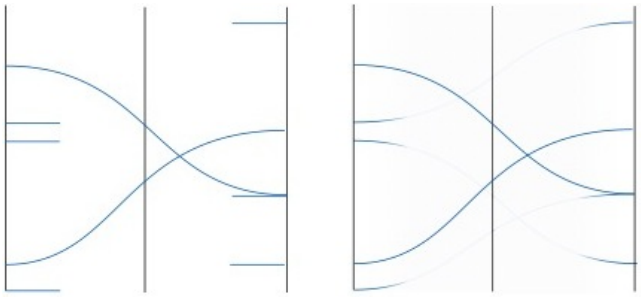


Figure 19: Two possibilities of visualizing gaps in the dataset. The stubs on the left indicate that the path continues, but there is no indication where it reconnects. On the right the path disappears behind the axis that does not contain a value.

3.3.3 *Drawing in-between Axes*

Several improvements have been made during recent years for drawing lines between axes. We experimented with several kinds and combinations of bundling techniques as well as with specific curves for supporting better recognition and improving aesthetics. Utilizing bundles of curves (Figure 18c) instead of straight lines visually smoothes the parallels coordinates and enables easier pattern recognition by tending a bundle lines to its center. Therefore bundles can be recognized and can be distinguished between each other much more easily than in common drawings.

As mentioned in the related work section, using curves instead of lines facilitates the tracking capabilities of a single data item across all axes, thus making it easier to distinguish between different paths. Usually, cubic and quadratic splines are combined to avoid oscillation above and below the vertical endings of the axes and to guarantee an aesthetic curvature.

We implemented polylines, a combination of quadratic and cubic splines and bundling for drawing lines, and showed them to users. However, they remarked that these approaches do not work well in combination with the extended areas, since they increase occlusion in the direct proximity of an extended area. Simple cubic functions

offer an alternative strategy for drawing lines. They are fully defined by two points and two derivatives on two successive axes. We found them particularly effective in the combination with extended areas if the curves between two axes intersect both axes in a right angle since this aids the overall legibility by creating a steadier path (Figure 18d and 17).

3.3.4 *List*

All the products are displayed in a list which is shown on the left side (Figure 17). To reinforce the visual metaphor of one path being one product, every path originates from the corresponding product entry within the list panel. The list can be understood as a special type of axis. However, some users also appreciate a familiar interface as a helpful start for dealing with parallel coordinates.

3.4 VISUAL QUERY GENERATION

How do people usually search for products? Often they start with some requirements they are sure about, some others that are not very specific and finally a few that are less or not at all important. They start searching for products that match these fuzzy requirements and put them into an imaginary subset. If no item appropriately matches their requirements, users then have to modify their search. On the other hand, if there are too many matches, they must narrow down the features that should be covered.

This strategy can be compared to an iterative process of sending queries to a database. Queries have to be relaxed if no results have been received, or new clauses are added to the query, or the existing ones are varied until the user is satisfied with the received subset.

The aim is to visually generate and refine a query, which specifies the user's requirements for the various attributes of a product. In the context of parallel coordinates this has to be expressed by selecting paths, for which several approaches can be combined: (1) The user can define a range on the axis with the help of two sliders. (2) Another possibly is to group items by brushing over certain paths. (3) We suggest to utilize the extended areas, which also work for histograms, for directly selecting certain attribute values. This mechanism is intended for axes of categorical and ordinal data. The sliders work equally well in all the mentioned kind of axes, thus it is possible to mix attribute range selection and direct attribute selection in a single query.

Early user feedback on these basic approaches eventually resulted in a consistent mechanism for choosing products rapidly and comfortably. Our prototype combines a slider-based interaction along with the possibility of selecting extended ar-

eas. Within an axis all selected attribute values and ranges are aggregated like an *OR* operation regardless of whether they have been selected by extended areas or by the sliders. The subset of selected products on each axis are intersected like an *AND* operation with all the other axes' selections. Products that fulfill the specified query are drawn in a salient color (Figure 20).

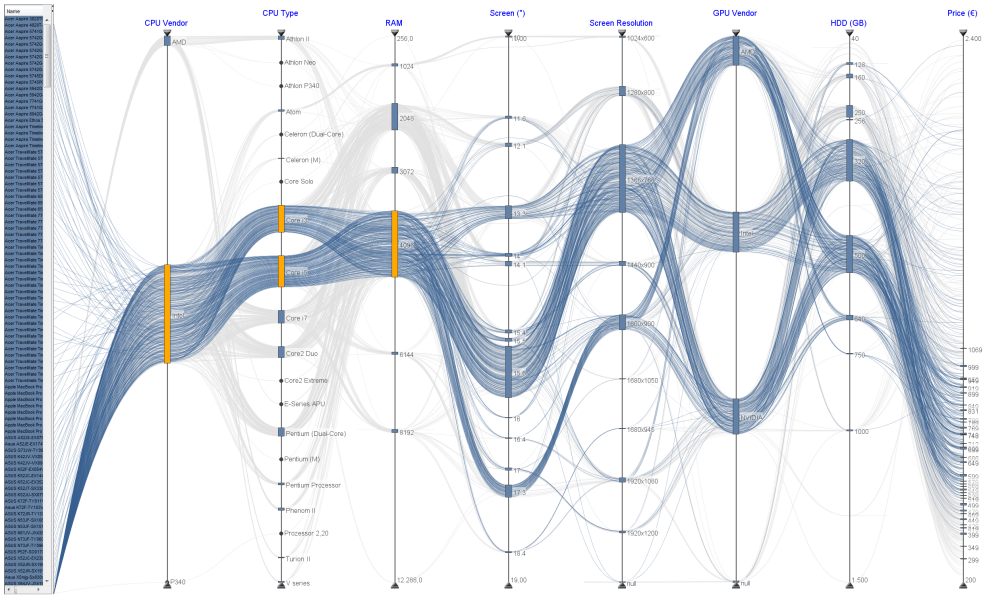
As mentioned before, all lines in the parallel coordinates display originate from the product list on the left. All products that match the entire query are at the top of the list, followed by partially satisfying products ordered by their rank and finally by the products that do not match up at all. All list items are drawn with the same color as the related paths. The list is reorganized instantly with respect to the product ranking depending on chosen ranges and values. The explicit marking of products is also possible (Figure 1).

In addition to the introduced selection mechanisms, our system provides a set of common parallel coordinates techniques that help users to organize the display. The axis direction can be switched from top down to bottom up. For comparing each axis with others, all axes can be horizontally dragged and dropped in arbitrary positions. The lines will instantly be redrawn during movement. Another capability allows users to zoom into the space between two axes with the mouse scroll wheel, making it possible to get a better overview of the path segments and their connection between two axes. As a result the other axes are temporarily squeezed together.

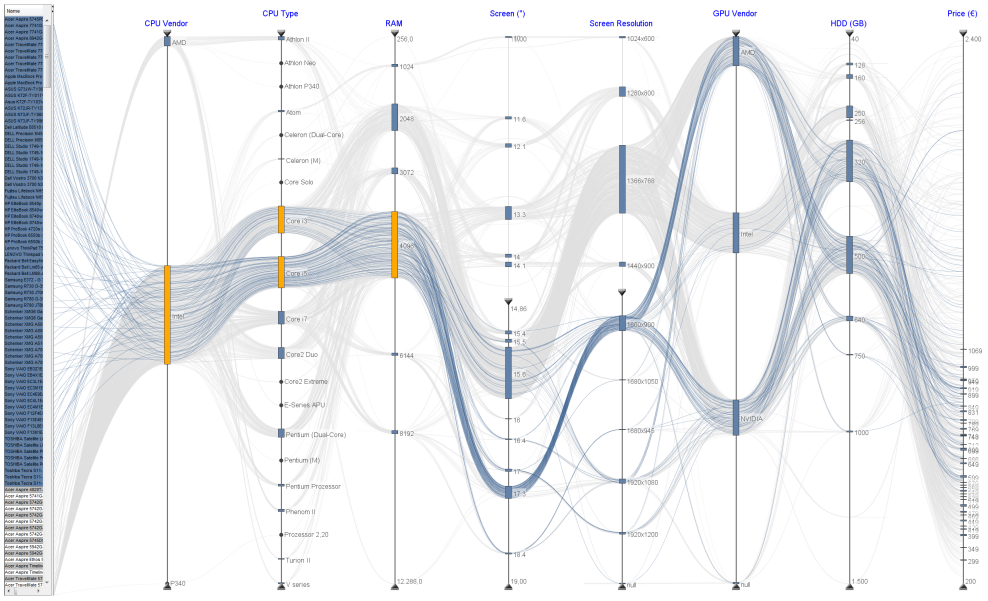
3.5 EXCLUSIVE DECISIONS

The ranges and sizes of displays people have nowadays are vast, including high resolution monitors, projection systems and smart phone displays. For making our Product Explorer available on these various platforms for different types of products, we have to particularly consider the number of axes that can be drawn at once since a useful drawing must provide appropriate space between axes so that the paths can be easily followed. As mentioned in the section about data issues, the axes' order should reflect the importance of the individual axis. Therefore, the most important axes should be preferably drawn first as long as sufficient gaps between axes can be provided.

For the remaining attributes, we propose an attribute repository, which helps the user to organize the axes that do not fit in the drawing area. With the repository, the user can decide to add an attribute to the drawing area or exchange axes between the repository and the drawing area. As depicted in Figure 22, the attributes can be easily chosen (via drag-and-drop) from the repository which contains only the hidden axes. The repository is implemented as a drawer and is only opened if necessary. The attributes themselves are depicted as miniature axes to maintain a similar aesthetics

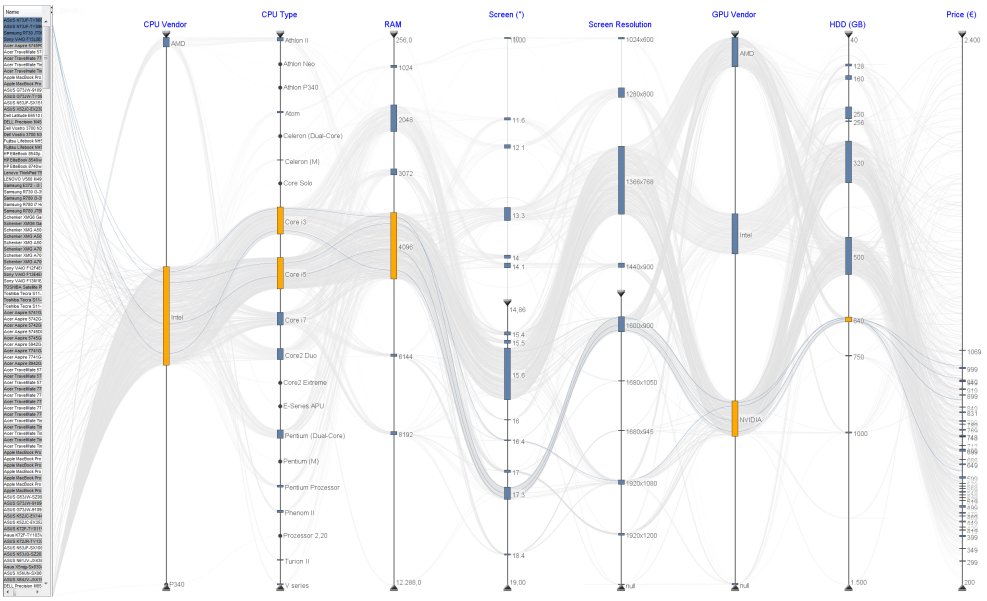


(a)

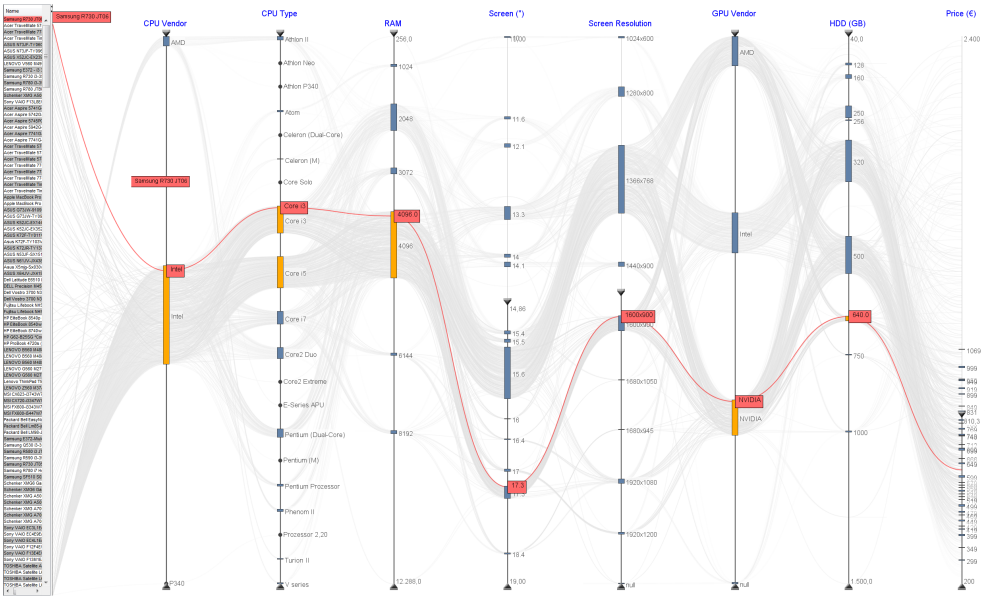


(b)

Figure 20: The visual query generation technique allows the users to express their requirements in a visual manner by selecting attribute values and by defining ranges. (a) A subset of the data is defined by selecting an Intel core i3 *OR* core i5 processor and 4 Gigabytes of RAM . All notebooks that match our constraints (and thus the query entirely) remain in blue, whilst the others are grayed out. (b) Defining ranges for a 15 inch or larger display with a high resolution further refines the query. See next steps on Figure 21.



(c)



(d)

Figure 21: (c) To finalize the query the attributes for a dedicated graphics card, and for 640 Gigabytes harddisk size are selected. Four notebooks remain and the price may guide our final decision (c and d). If necessary, e.g. if the result set is empty or the remaining products are not desired, the user is able to relax or to adjust the criteria on each axis, either by deselecting an area, by adding another area or by moving the range sliders.

and to aid the user when dealing with attributes that are almost as important as the previously selected ones.

Moreover, we propose an exclusive decisions technique to reduce the number of axes in the drawing area. Additionally, this technique supports the process of fast decision-making that our prototype is intended for. We assume that according to the individual importance of each axis, the major attributes will be chosen first and the user is very certain about these decisions. Therefore, there is no need to waste space by continuing to draw such axes. After selecting an extended area or adjusting a range on an axis, the axis will disappear and thus all lines that do not belong to the current selection will be hidden. However, sometimes it may be necessary to step back and to change a decision already made.

Therefore, a history below the parallel coordinates display shows the decisions (the chosen axes AND its chosen values) the user has had already made and they are then able to reselect an axis that needs to be taken back into consideration. All these techniques are facilitated by appropriate animated transitions, which help the users in recognizing the appearance and disappearance of axes and paths.

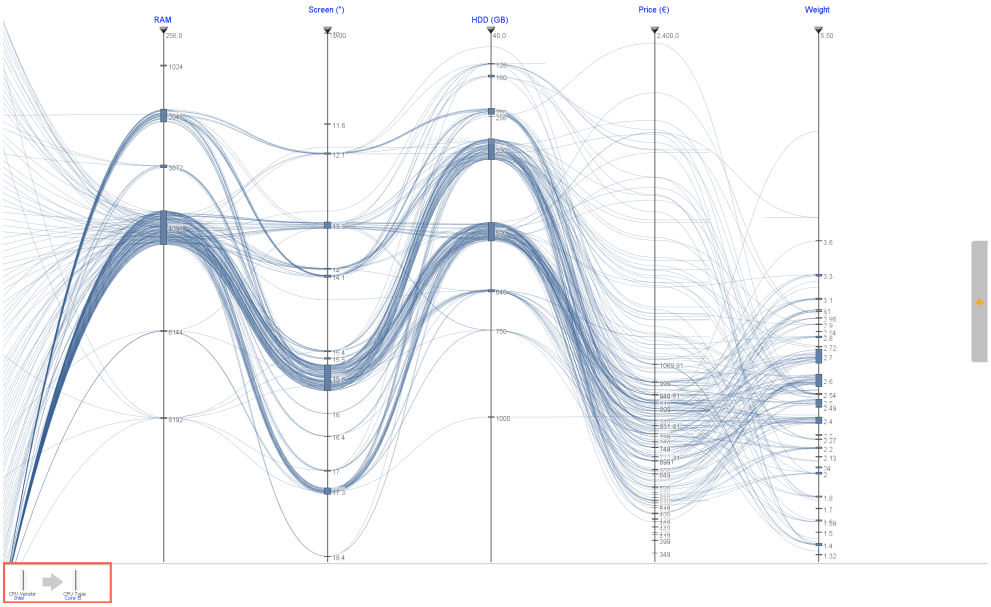
Unfortunately, because of the general orientation of the parallel coordinates, scalability issues with the vertical space cannot be solved with the same methods for the horizontal direction. Our aim is to depict all products at once, so pre-computing techniques like binning or clustering are not suitable for this purpose. Instead, we expect that a Focus+Context technique along the axes might be useful.

3.6 USER FEEDBACK

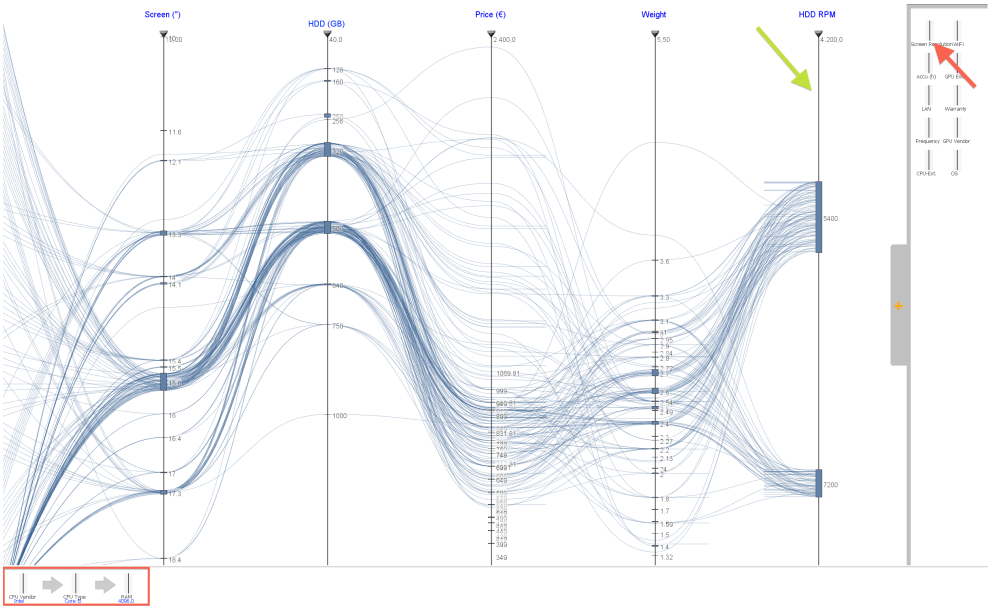
Our application prototype is implemented in Java and we employed Java2D for drawing operations. Java2D provides convenient functions for drawing curves and the graphics update rates are still smooth on high resolution displays.

We performed an initial user study to assesses the basic usability of our implementation with a fixed set of axes and without the axis repository on the bottom of the display against a typical Web interface consisting of entry fields, drop boxes and option buttons and containing a submit button, which invokes the search by requesting a result page.

In particular, we were interested in the following aspects: (1) How does the representation of a product by a single path across all the axes work for the participants? (2) Is an extended area still recognized as a single point or attribute? (3) Which possibilities for drawing axes and lines are generally preferred? (4) Does the combination of different kinds of selection techniques work well? (5) Is the Product Explorer appreciated by the participants or do they prefer a classic web interface.

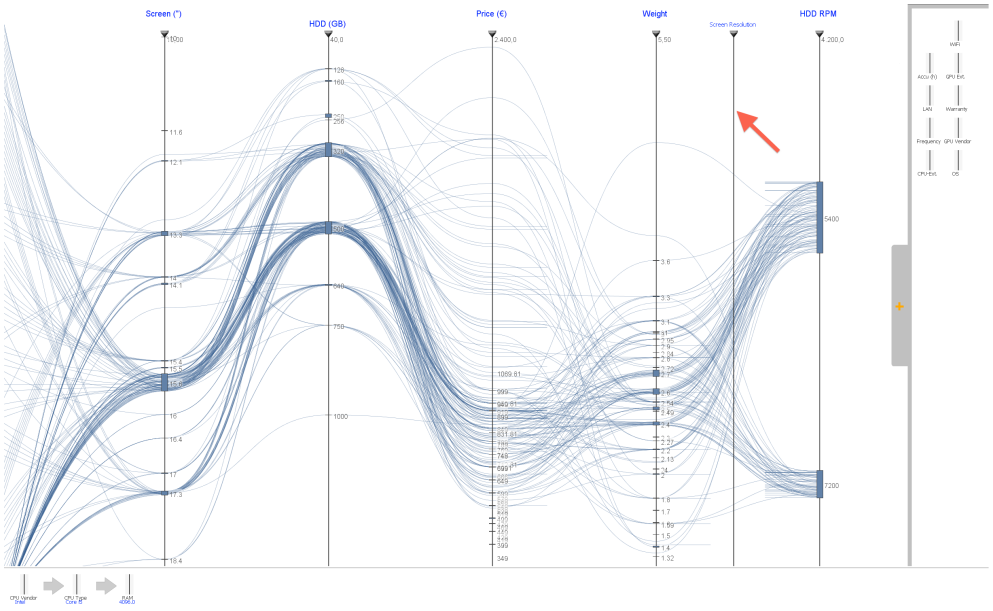


(a)

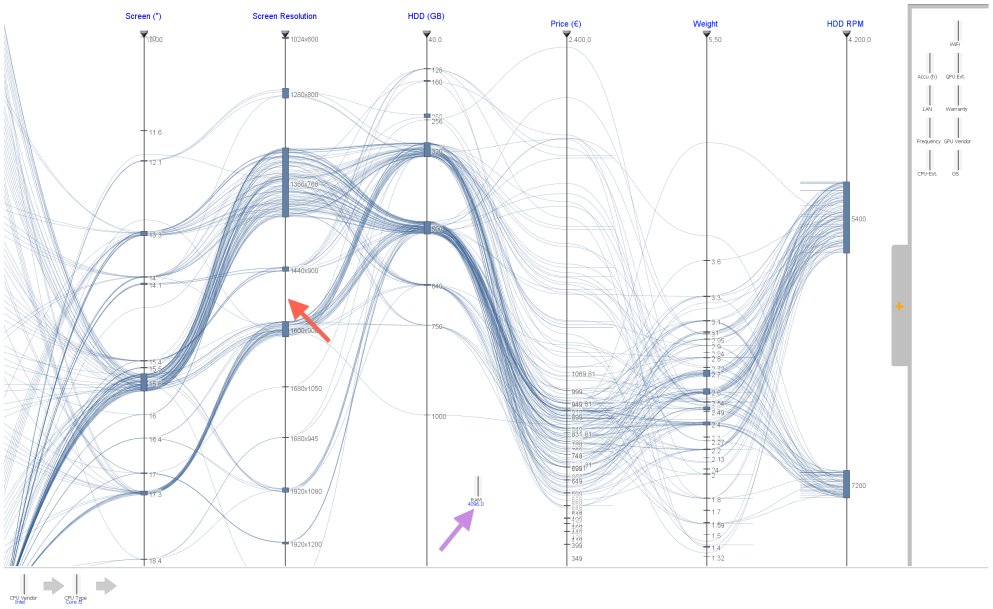


(b)

Figure 22: Exclusive decisions and the attribute repository: Only the most important axes are drawn at first. If the user is certain about a particular value for an attribute (e.g. Core i5; see (a) and (b), red rectangle), they can remove an axis by choosing an extended area of a particular axis (a, black pointer) or by selecting a range. There is no need to display this particular axis any longer and it disappears together with all paths that do not fulfill this decision. All previously chosen axes along with the selected values are depicted in the decision panel below ((a) and (b), red rectangles), so users can track the decisions they made for certain attributes. By making decisions attribute by attribute, every step releases some horizontal space which is automatically filled with newly appearing axes ((b), green pointer) that were originally of less interest but could be now important for a final decision ((b), green pointer). The axes of less importance are kept as miniatures in the attribute repository on the right. See next steps on Figure 23.



(c)



(d)

Figure 23: The miniatures can be dragged to the main drawing area and dropped at an arbitrary position where it then transforms to a regular axis which also becomes connected instantly((b),(c),(d) red pointer). The positions of all axes are readjusted according to the number of axes and the available space. If it is necessary to rethink a decision the user can drag one or more axes back into the main display area (d, purple pointer).

None of the eight participants were familiar with our work. The study comprised six test product assignments that were selected for variety and with two different difficulties (three each). For example:

- EASY Look for a notebook under 1000 Euros, which is as light as possible and has an AMD processor.
- DIFFICULT You want a notebook with an Intel CPU and an Intel graphics chip. The CPU should be a Core i3 or i5. Find the fastest notebook with the lowest price.

After a brief introduction to the parallel coordinates interface, the participants were asked to search for and to select the most suitable product(s) for the given assignments. To minimize learning influences and other effects, the participants were divided into two subgroups: one subgroup began the trial with the Product Explorer and the other group started with the Web interface. Subsequently, the participants were asked to answer a questionnaire. It consisted of two parts. Part one asked about the interaction in general (e.g. how natural it was to use, preference for an interface etc.). The second one was designed to gather feedback about the aesthetics decisions that have been made during development. Rating questions used a Likert scale from one to six.

A *t*-test was conducted to compare the results of both interfaces. We used an alpha level of .01 for all statistical tests. As expected, the Product Explorer was significantly faster for the difficult tasks (M 39.12, SE 9.14) vs. (M 92.00, SE 9.14), $t7$ 6.161, $p < .001$ as well as for the easy tasks (M 32.20, SE 4.19) vs. (M 72.00, SE 3.83), $t7$ 8.67, $p < .001$ (Figure 24). The subjective ratings confirm these results. Overall, the participants prefer the Product Explorer over the Web interface; (M 5.50, SE .19) vs. (M 3.25, SE .16), $t7$ 13.74, $p < .001$, which reveals a high acceptance for our interface that uses novel interaction techniques and visual metaphors that none of the participants had ever seen before.

The “one path is one product” dualism seems to be very comprehensible for the users with a rating of 5.8 as well as the “one extended area represents one value” metaphor with a rating of 6. The range slider was judged as helpful by nearly half of the participants, but the area selection technique was considered much more effective. The tutor’s observations during the assignments confirmed the predominant usage of the area selection.

It might be that Grahams method [35] provides an aesthetic curvature for a single path. However, for the visual impression of the entire plot, the users favor the simple cubic curve approach (7 of 8 participants) which they found to be more steady and legible. Furthermore, the participants preferred the extended area instead of histograms (6 against 2) if they were forced to choose between the two. Moreover, those who voted for the histogram would actually like to see a combination of both.

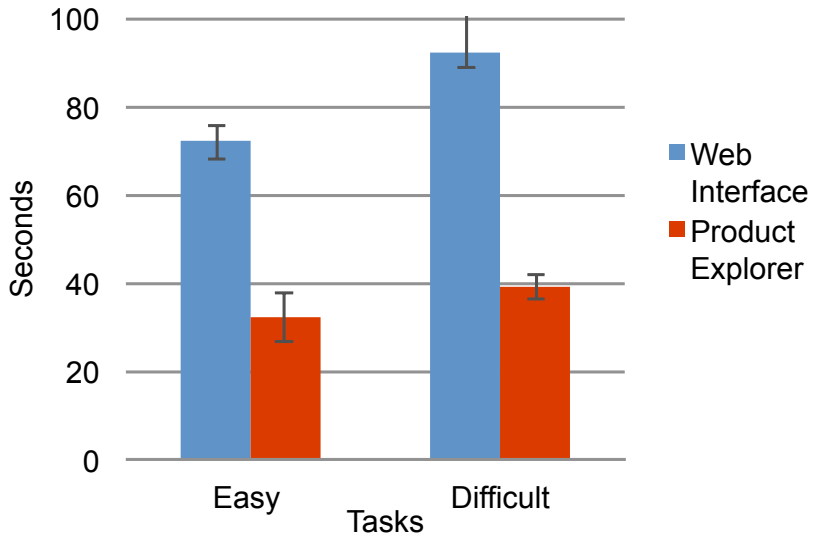


Figure 24: Mean task completion times over all participants for both levels of difficulty along with their standard errors.

The participants of our study suggested several features to improve the prototype. The most frequently mentioned ones were: (1) to reconsider the ordering of the values along the axis (most wanted by frequency) and (2) to start with only a few axes and to slide in additional axes on demand. In light of this feedback, the latter feature has already been implemented and discussed in this paper.

Not surprisingly, all of the participants of our study would like to use our Product Explorer for choosing products on the web. This was also the general opinion of people who used our system during public demonstrations. The most interesting finding during these events was that other researchers, such as engineers, architects and archaeologists, are generally not familiar with parallel coordinates. Nevertheless, they are faced with similar problems (e.g. finding the best-suited material for a bridge, searching for a glass with certain requirements or tagging and organizing thousands of small archaeological pieces, which have numerous attributes). On the whole, after a brief introduction, they are excited about how helpful such a tool can be for their own field of research. It can help them to organize, present (especially with capabilities to mix categorical and continuous data) and find items by visually generating a query.

Our user study is not particularly fair regarding the most advanced web shop search interfaces, which already utilize all new capabilities that are given by the so called "WEB 2.0". We believe that a web form made of sliders, ranges and also checkboxes which realizes an instant approach to display the result list by any change of the user's requirements and without reloading the entire website (for example using the XML-HTTP request) can be nearly as fast as the Product Explorer for certain queries.

But even with these improvements a list-based results view cannot show all items at once, it cannot display whether there are other possibilities in the vicinity of a user's requirements that do not exactly match. On the contrary, with the Product Explorer, the user can see what configurations are offered by the market, which influences further expectations. Moreover, if the users' wishes cannot be fulfilled, they will be able to see why and what alternative options are available. Additionally, before users make their next decision, they can anticipate what options will be available.

3.7 PERSONAL MARKET ANALYSIS

As in business, most private purchases also start with some kind of market research to learn what products are currently offered and what combinations of features exist. This personal market analysis influences our early expectations of a product and results in our final product requirements. Thus, our tool additionally provides capabilities to reveal important aspects regarding market research tasks, e.g. displaying market distributions along a certain axis, comparing shares of different vendors who provide a similar component or identifying relations between adjacent axes and tendencies across several axes (see Figure 26).

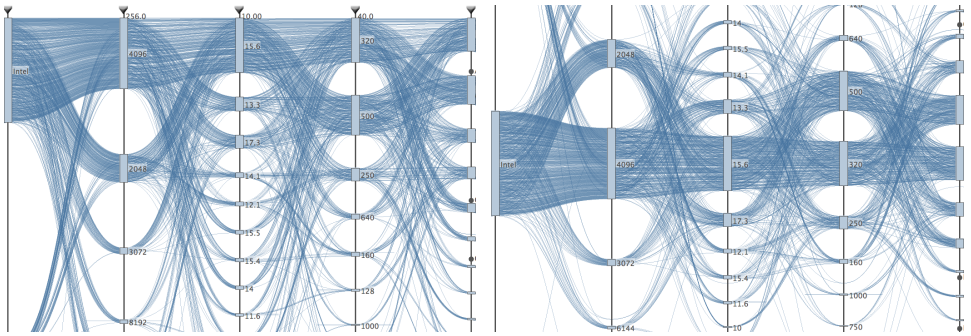
With that goal in mind it can be useful to rearrange the order of data points along an axis by manually dragging them to an appropriate position or by automatically assigning a position (e.g. top-down by occurrence). Figure 25 shows an example scenario. For categorical and ordinal data the most frequently occurring attribute value would be shown on top of an axis. Even for pseudo-continuous data this approach can be helpful, if only a few values exist along the continues axis (e.g. hard disk velocity), which has clearly some potential for optimizing the layout of the parallel coordinates display to increase its expressiveness.

REGULAR ORDERING The regular ordering serves as a reference. It uses the implicit ordering of continuous and ordinal attributes and alphabetical ordering of categorical axes.

TOP DOWN The top-down method (bottom up is also possible) follows the idea that values that occur more often within a data set could be more important and should be placed prominently. Thus, the attribute values of categorical and ordinal axes are ordered by increasing frequency. Truly continuous axes such as the price are ordered by increasing or decreasing value.

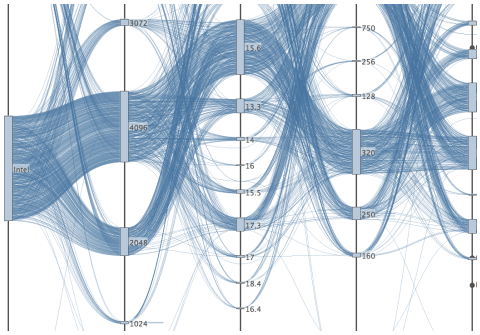
CENTER SPREAD The center spread ordering is similar to the aforementioned, whereas the most frequent value is assigned in the vertical center of the axis and subsequent extended areas are placed alternately above and below with decreasing frequency.

CROSSING MINIMIZATION This approach is inspired by a classic algorithm for drawing layered graphs which was suggested by Sugiyama [107] for his barycenter-based layer-by-layer sweep. A parallel coordinates plot with its axes and extended areas can be considered as a type of layered graph. We implement a barycenter-heuristic which calculates a permutation of extended areas of the current axis depending on the barycenter of the previous position (at the previous axis) of all lines belonging to a particular area.



(a) Top down

(b) Center spread



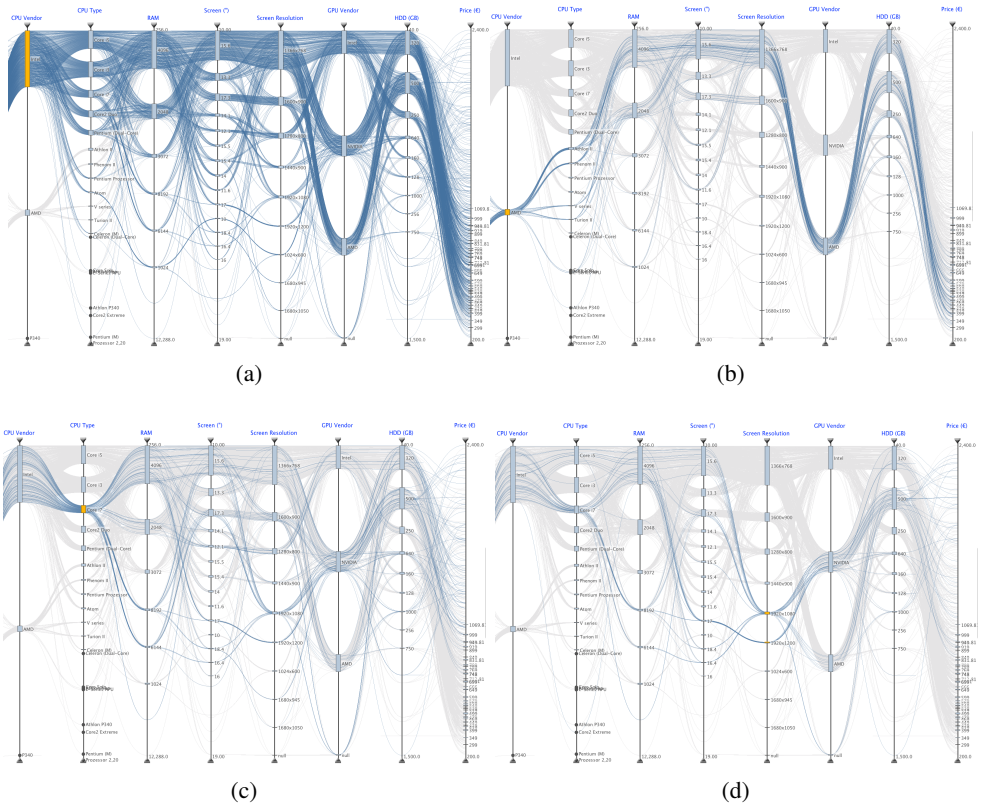
(c) Crossing minimization

Figure 25: Layout methods: The center-spread layout clearly show the most often used product configurations and their variations. With the crossing-minimization layout product sets with similar properties seem more difficult to trace.

For some continuous data attributes this might not be an option, however, most often a quantization of continuous data could also be treated in the same way, e.g. for the weight attribute in our notebook data set. Categories might be introduced such as less than two pounds, two to three pounds, three to four pounds, and so on.

3.8 CONCLUSIONS AND FUTURE WORK

Our work on the Product Explorer and in particular our pilot study clearly confirm that parallel coordinates with a number of crucial improvements and enhancements



(a) (b)

(c) (d)

Figure 26: The Product Explorer as a tool for market analysis reveals some interesting patterns in the notebook dataset. Here we show the TOP-DOWN APPROACH which assigns positions to the extended areas depending on how many paths cross an area, for example to assess market shares of different vendors. The most common configuration in the market can be seen immediately on top of the plot. Other configurations can be easily investigated by using the visual query generation technique. For example: (a) Notebooks that contain an Intel processor are distributed over the entire price range. (b) Notebooks with AMD processors tend to have lower prices and the built-in screens usually do not have a very high-resolution. (c) Core i7 models usually are more expensive, but the performance of the built-in components varies and and the components are not always high-end. These results are somewhat unexpected. (d) The screens with high-resolutions are bigger and mostly bundled with dedicated graphics and Intel processors (i7, i5, single i3), but often, harddisks of lower capacity are integrated. Surprisingly, these models are not the most expensive ones.

can be successfully used to implement product search interfaces on the Web that are intuitive, fast and provide a user with much more information than the commonly used text-based interfaces. The following layout decisions and enhancements contributed significantly to the usability of our tool:

- Extended areas instead of histograms or even points facilitate the tracing of product attributes.
- Cubic curves in combination with extended areas lead to a tidier display of a large number of products by avoiding occlusions near an axis.
- An intuitive visual interface based on a set of simple rules is the key for quickly narrowing down the product search.
- The attribute repository is important for the scalability to various display sizes. Also, this way one may start with the most important axes.
- The decision bar reduces the complexity of the display by reducing the number of considered attributes and products, which have already been considered.

There are many further possibilities to refine and optimize our tool. Missing attribute data is currently only treated in the visualization, but it also needs to be handled during the visual query specification by adding, for example, a selectable pseudo-value or pseudo-category on each axis. Support for multiple range specifications on a single axis might be useful when working with data sets containing mainly continuous axes. We also see some potential for reducing the overall number of axes by merging related axes, for example, more specific categories could be generated by combining CPU vendor and CPU type or hard disk size and hard disk velocity.

People working in various companies asked for a version of our Product Explorer prototype to present and search their own product catalog. Additionally, and perhaps of greater interest, they would like to visualize their respective markets with a comparison of all products from the competing companies. Thus, our tool needs to support different product types and their relationships and dependencies.

Our most important next step is the development of an HTML-only implementation without the need for plugins. We plan to integrate this new version in a web shop interface or a product search site to gain more experience with a larger audience and further advance the visual Product Explorer interface.

Part 4

VISUAL ASSESSMENT OF ALLEGED PLAGIARISM CASES

We developed a visual analysis tool to support the verification, assessment, and presentation of alleged cases of plagiarism. The analysis of a suspicious document typically results in a compilation of categorized "finding spots". The categorization reveals the way in which the suspicious text fragment was created from the source, e.g. by obfuscation, translation, or by shake and paste. We provide a three-level approach for exploring the finding spots in context. The overview shows the relationship of the entire suspicious document to the set of source documents. A glyph-based view reveals the structural and textual differences and similarities of a set of finding spots and their corresponding source text fragments. For further analysis and editing of the finding spot's assessment, the actual text fragments can be embedded side-by-side in the diffline view. The different views are tied together by versatile navigation and selection operations. Our expert reviewers confirm that our tool provides a significant improvement over existing static visualizations for assessing plagiarism cases.

4.1 INTRODUCTION

Text reuse is ubiquitous and ever-present. News messages travel from website to website with only slight changes in wording or identical text fragments emerge in a passed version of a bill that have previously been released in documents drawn up by lobbyist groups. While these cases often have little or no consequences for the plagiarizing authors, this is different for student essays or PhD theses accused of plagiarism. In these cases, text passages originating from other authors have been either directly copied or slightly rewritten without properly referring to the original sources which is, in the best case scenario, a lack of scientific thoroughness. Claiming that a given piece of writing has been plagiarized can have severe consequences for those accused. The supporting evidence of such an accusation needs to be presented in a convincing way or it may be refuted, regardless of truth. To ameliorate the situation,

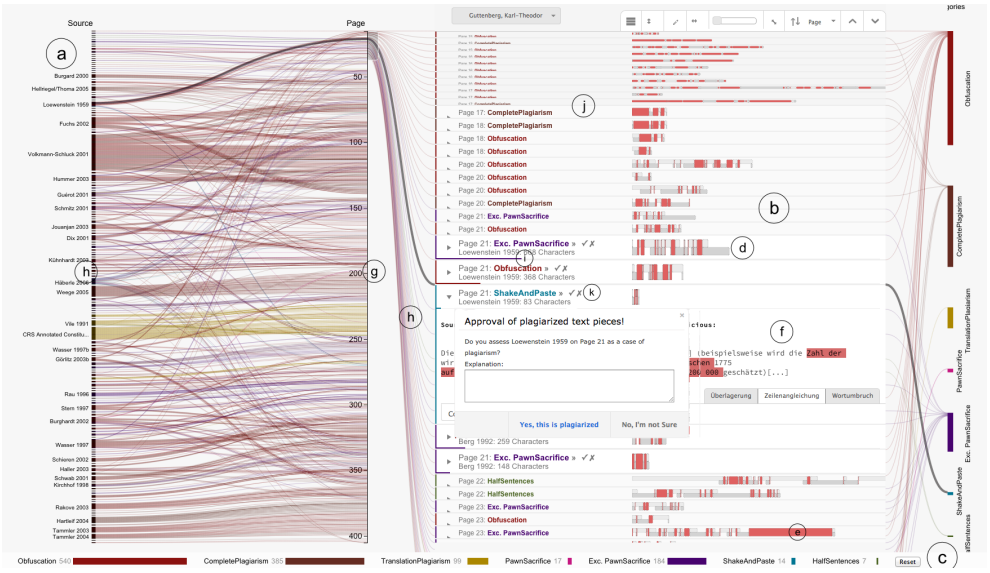


Figure 27: Our visual tool for assessing cases of plagiarism displays the types of plagiarism found on the bottom (c), a list of difflines (b) (glyph-based visualization of the finding spots (d)) in the center, and an overview on the left (a). Each finding spot is encoded as a single diffline. Copy-and-pasted passages are marked in red (e). A finding spot can be opened for a side-by-side comparison of suspicious and original text fragments. The overview reveals the distribution of finding spots across the document (g) and their relationship to the sources (h). The overview supports brushing and selection to define a subset of finding spots to be displayed in the diffline view.

we developed an interactive visual analysis tool (Figure 27) which provides effective views and appropriate linking and filtering techniques to explore an alleged case of plagiarism from the entire document down to individual suspicious sections of text (finding spots). An overview provides insight into the distribution of finding spots across the document, their lengths and categorizations, and their relation to sources

and authors. Effective filtering, linking, and navigation techniques facilitate the process of focusing on different aspects of the case, such as a certain source, plagiarism category, or the largest finding spots. The selected finding spots are presented as a list of difflines, a glyph-based abstraction for revealing the inner structure of a finding spot. They serve as intermediate representation between overviews and actual text by encoding the modifications that turned the source text fragment into a finding spot by explicitly highlighting the copy-and-paste sequences. For drilling down a finding spot, the actual text fragments can be opened below as textual view. Therefore, along with our set of expert functions, each finding spot can be considered in detail and, if needed, be reassessed or altered and, eventually, the assessor must approve or reject it from the list of suspicious fragments.

The specific motivation for this work stems from our professional experience as developers of the text reuse search engine Picapica [82] and as initiators and organizers of an annual international competition on plagiarism detection, called PAN [83]. In this context, we are also in contact with experts and members of the German anti-plagiarizing community. Discussions with our colleagues and experts, as well as a review of available tools, revealed that most plagiarism search engines present their results as running text containing color-highlighted word sequences at positions where text has been reused whereas different colors hint at different source documents. A few tools provide basic overviews with only very limited interaction capabilities. Such solutions may suffice if short texts have to be analyzed. However, they do not scale gracefully with text length, nor with complexity of a plagiarism case. In such cases, a lot of information concerning different aspects of work and practice needs to be considered by experts or discussed in a council charged to audit a suspicious case. Besides answering questions about the overall characteristics of the suspicious document, the assessment of each individual finding spot remains crucial. The aforementioned solutions mostly provide scrollable page-based textual views for browsing the entire document. However, scrolling a 400 page document interrupted by reading and comparing each finding spot to the related source is a tedious task.

The central contributions of our plagiarism analysis tool include a three-tiered approach for exploring alleged cases of plagiarism, new overview paradigms for navigating and selecting subsets in a suspicious document, difflines as effective glyph-based abstractions of differences and similarities between two text fragments, and the support for fluid and coherent interaction between the different levels of detail. As an initial data set, we chose the most elaborate collections of suspicious PhD theses, GuttenPlag [70] and VroniPlag [71]. Reviews with our plagiarism experts confirm that our tool can effectively support their workflow and provides a significant improvement over existing static visualizations for assessing plagiarism cases, especially regarding time savings during the assessment process and in visually supporting councils and committees in forming an opinion about a plagiarism case.

4.2 ANTI-PLAGIARISM COMMUNITY

In Germany, a very active and self-organized anti-plagiarism community is committed to finding and documenting cases of plagiarism in PhD theses. The members document their results in public wikis such as GuttenPlag [70] and VroniPlag [71]. They scrutinize documents that have been suspicious to one or several members for various reasons. It is an ongoing process which typically takes months or even years since all community members are volunteers. Each finding spot is documented, compared with the work it was allegedly taken from, and published with specific information such as position in the suspicious document, position within the original document, original author, etc. A single source or even multiple documents of the same author(s) are often used repeatedly. Eventually, the finding spots are categorized as different types of plagiarism. The most common categories defined by the community are described below, as are their colors used in our system.

- (ALMOST) COMPLETE PLAGIARISM: a section largely produced by copy-and-paste.
- OBFUSCATION: a text passage which is more or less paraphrased, often by simply substituting words with synonyms or inserting/deleting select words here and there.
- PAWN SACRIFICE: text from a cited source is used but is referred to somewhere else in the document.
- EXACERBATED PAWN SACRIFICE: text is copied straight from a source and a correct reference is cited, but the reference is introduced with "'likewise ...'" suggesting that there is a similar statement but not equal text.
- SHAKE AND PASTE: longer text sections, typically paragraphs, are taken and mixed from different sources.
- HALF SENTENCE MENDING: short sentences or sentence fragments from a source have been used.
- TRANSLATION PLAGIARISM: a text translated from a foreign language source which was more or less rephrased.

The barcode visualization [71] is the most common visualization utilized by members of the anti-plagiarism community. It provides an overview of a suspicious document and is used to demonstrate the current status of an ongoing investigation. The horizontal barcode shows the pages of a document as vertical stripes which indicate whether one or more finding spots occur on a particular page. A five-level color scale defines the amount of suspicious fragments per page.

The depiction is usually just a static image, but some can show the detection of finding spots over time in an animation. Another non-interactive visualization [114] of

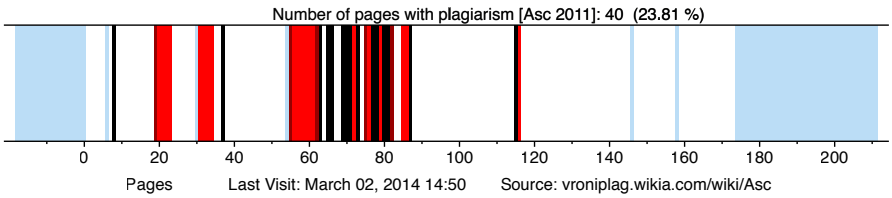


Figure 28: The barcode visualization example from the VroniPlag Wiki [71] shows the current status of the investigation. Not to be confused with with our colors of the different categories.

the GuttenPlag community employs a page-based view of the entire document. Each finding spot is shown in a color that corresponds to the author of the source document. However, in Guttenberg's case, with nearly 400 pages and 138 different authors, the colors are too similar to allow an unambiguous assignment to an original author. Nevertheless, such a visualization provides a solid overview of the amount of text taken from others and it works well for minimal sources.

4.3 RELATED WORK

A different kind of text reuse, documented by the Lobbyplag website [75], reveals changes in regulation drafts for the General Data Protection Regulation (GDPR) of the European Union. It allows a comparison of changes within the committee amendments and relates them to lobby proposals about the same topic that might contain similar content and wording. A horizontal barcode spanning the entire page serves as overview and navigation tool. The amendments and lobby proposals are shown as a side-by-side comparison without visually linking the texts in any manner. Only text changed in the amendments and proposals is highlighted in red (removed) and in green (newly inserted).

In her book, Weber-Wulf [126] gives an overview of the current situation of plagiarism and its detection. More than 50 plagiarism detection systems can be found, some offered as commercial products such as Turnitin [51]), Ephorus (now merged with Turnitin), and Urkund [87], and some merely small open source tools. Since 2004, almost all of the available systems have been repeatedly evaluated with respect to their detection quality and fitness for purpose, the results of which have been published at [125]. Since 2008 these evaluations also assess usability. In this regard, few systems achieve more than 70% of the available points (both on an objective and a subjective scale), so that most are rated "poor" or even "unacceptable" [127]. We surveyed the available systems with regard to their visualizations employed: none of the systems individually visualize findings and only few provide abstract overviews of their findings, which usually boil down to tables that give numbers of findings alongside document names.

Gipp and Meuschke [32] developed a visualization based on an underlying citation-based plagiarism detection algorithm. The documents are also arranged side-by-side with overview bars in-between representing the entire document. References are shown as dots in each overview bar and identical citations are connected by a curved line (see Citeplag website [96] for examples). They also published an interesting survey about the state of the art in detecting academic plagiarism [66]. The paper of Jänicke [52] offers several visualizations of textual differences and commonalities of different English Bible translations, such as Text Re-use Grid, text-centered visualizations, and Sentence Alignment Flows, which strongly resemble the Wordgraph metaphor described in Part 1, but was already published in 2011 (see Publication a).

The visualization of regular diff algorithms is also related to the depiction of plagiarism. Windiff [68], an older tool for comparing different revisions of source code, provides vertical bars beside the text views which show differences of code revisions by coloring variations and identifying moved parts. Contrary to our approach, it does not focus on the equal parts by particularly aligning the changed parts alongside the remaining ones. Unfortunately, this approach is barely applicable to continuous text that is not explicitly wrapped, such as source code. Chevalier et al. [14] propose a different approach by utilizing an animation technique for smooth transitions between text revisions. Animations could also be useful in the case of plagiarism for speculating how the author created documents from a set of sources.

Another topic related to certain aspects of our approach is the visual tracking of changes made during consecutive revisions or edits of single text documents, which is exemplified in HistoryFlow by Viegas [119], the Wikidashboard by Suh [108], or the Chromogram by Wattenberg [124]. However, here the focus is on the continuous evolution of a single document instead of revealing the relationship between multiple source documents and a single suspicious document as in the case of plagiarism. An interesting approach, also supporting the navigation between several levels of abstraction while exploring large texts, was provided by Koch [57].

4.4 DESIGN PROCESS AND VISUAL CONCEPT

The annual PAN [83] competition on plagiarism detection, which we organize, and our own text reuse search engine Picapica [82] focus on the automatic retrieval of plagiarism. Visualization was not a necessity when plagiarism detectors were evaluated in the past (see for example [85]). Nevertheless, participants of the competition and customers of our Picapica service alike frequently ask for solutions that save work time when reviewing plagiarism cases.

The development of Picapica, as well as that of our first visualizations, was advised by the German anti-plagiarism community. Their process of *manually* analyzing a

suspicious PhD thesis can be summarized as follows: after a suspicion has been raised, the document in question is scanned for further dubious text spots, usually by manual retrieval. For each so-called finding spot, a corresponding text fragment from a potential source document is listed. In addition, the finding spots are classified with respect to the perceived way in which the suspicious text fragment has been derived from its source (e.g., by obfuscation, mending the sentence fragments of the original, or simply by copying and pasting).

The rationale for identifying as many finding spots in a suspicious document as possible is due to the fact, that, in practice, a single, short plagiarized text passage is considered insufficient evidence to make a case against the document's author: for example, the author might claim a mishap. Therefore, a complete analysis of a suspicious document is a strict necessity to support and defend plagiarism allegations. For instance, when councils need to form an opinion about a plagiarism case, a lot of information concerning different aspects about work and practice needs to be considered by experts or discussed in the council. Based on the identified finding spots, they have to answer those questions which are critical to a thorough assessment of an entire suspicious document: How are the finding spots distributed among the pages of the entire document? Which categories of plagiarism are present in the document and which of them are most frequent? How many sources were used? Which sources are used most for paraphrasing text and to what extent? Which sources appear in which category and how often? What is the average length of the finding spots or, more specifically, what is the distribution of their lengths? Besides the consideration of these general characteristics of the suspicious document, the assessment and reassessment, presentation, and discussion of individual finding spots is an important part of the process.

For an effective support of this process and to answer the aforementioned questions in a convincing way, we derived the following key elements of our visualization system:

- An overview is needed to support group decision processes in order to gain insight into the distribution of finding spots across the document, their lengths and categorizations, and their relation to sources and authors.
- The most important requirement, saving time in forming an opinion about a list of finding spots, is facilitated by introducing a compact glyph-based representation which demonstrates the relationship between source text and finding spot. This intermediate representation visually emphasizes the copy-and-paste fragments of a finding spot and therefore simplifies reaching a consensus about a finding spot without looking at the text.
- The actual text fragments—source text and finding spot—are sometimes still necessary and can be opened below a diffline as a side-by-side or merged view.

- Effective filtering techniques facilitate the process of focusing on different aspects of the case, such as a certain source, plagiarism category, or the largest finding spots in order to verify the claim of plagiarism or to convince council members with respect to a given case.

Our visual plagiarism analysis tool is aimed at people who typically do not have any experience in advanced information visualization and need to focus on the analytical task. Thus it is clearly structured and only consists of the category view on the bottom, the overview on the left, and the main view in the center which shows the list of finding spots visualized as difflines. The overview and the main view are linked and we provide appropriate navigation techniques to explore the entire document down to individual finding spots.

4.4.1 *Visualizing All Finding Spots at Once*

The overview visualizations enable users to interactively explore different aspects of the structure of the suspicious document. Our graph-like view relates the pages where finding spots occur and the extent of finding spots, as well as the different finding spot categories to the source documents from which they were allegedly taken. The overviews are exchanged according to the overall sorting order (by page within suspicious document, by text length of the finding spot, by plagiarism category, or by source document, see Figure 29). A crossing minimization is applied to improve the aesthetics.

The individual overviews also allow the users to navigate the entire document and to filter the finding spots based on the aforementioned features. For continuous features, a range-based filter is provided: a particular subset of the pages or a set of really short finding spots can be selected. Discrete values are filtered by directly selecting their visual representations. Filtering of finding spots works consistently across all views and defines the finding spots that are contained in the diffline list. The currently viewable detail of the list is emphasized. The finding spots reveal their position or ranges within the overview by connecting the vertical positions to the respective finding spot entries with paths crossing the gap between both views (see Figure 27(h)). The existence and controls of these paths are being adjusted on the fly while scrolling, filtering, or reordering the list, so it is always clear which subsets (in the categorical/-source views) or which ranges (in the page/length view) of difflines can be seen at the moment. The category bar shown at the bottom provides information about the different kinds of plagiarism and their respective numbers occurring in the document under investigation. It also allows the selection of a subset of categories. Additionally, if enough horizontal space is available, the category bar can be integrated into the right-hand side by connecting the finding spots with their categories.

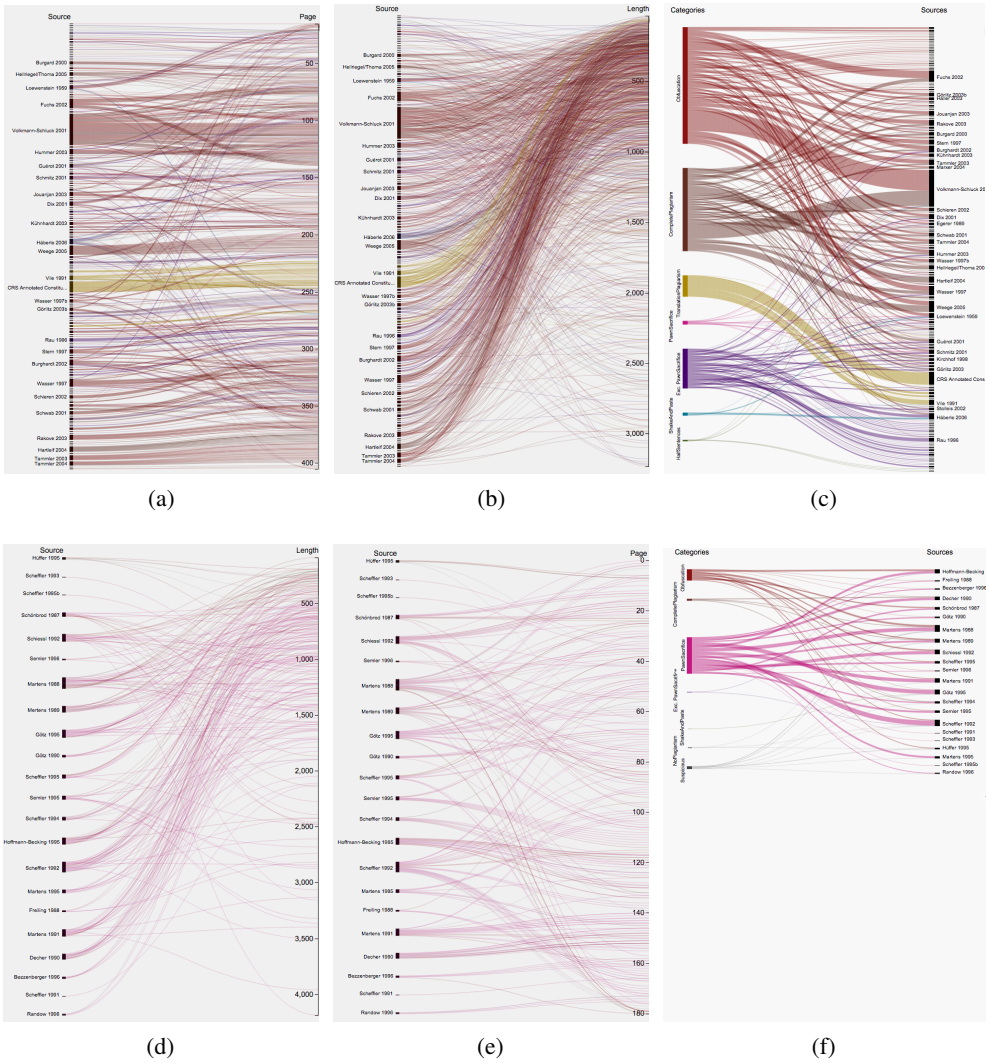


Figure 29: Overviews provided by our system. (a,d) The page view shows the entire document and gives an overview of the distribution of finding spots along the pages on the left side and the relating source documents on the right side. (b,e) The length view gives an overview about the length distribution of finding spots. (c,f) The category view visually assigns the particular kinds of plagiarism on the left to the source documents. In each view, the amount of text taken from various source documents can be visually estimated and compared on the right side.

4.4.2 *Finding Spots and Difflines*

The finding spot entries with their difflines are arranged in a tabular layout within the main view in order to enable the comparison of plagiarism patterns of several finding spots. Each finding spot is represented as a horizontal entry in which all of

its essential information is shown (Figure 32). Our central goal when designing the diffline was to visually convey information about the structure of the finding spot and its differences to the source without being forced to read the text itself. Our analysis of the finding spots of available cases revealed that, across all the different plagiarism categories (except translation plagiarism), there is a lot of direct copy-and-paste occurring. The frequencies and patterns seemed somewhat different, but it was difficult to judge by solely comparing two text fragments side-by-side. This observation led to the idea to provide a visual diff representation that expresses how a text changed between a finding spot and its source. With an appropriate glyph alphabet we are able to present the changes in a visual manner:

1. Identical fragments (copy and paste)
2. Modification of text fragment resulting in fewer, equal, or more characters
3. Insertion or removal of characters at a certain position (boundary cases of the above)

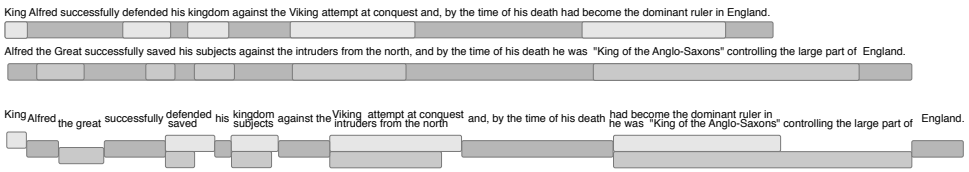


Figure 30: (Top) Two lines of similar text with only slight differences in wording. Identical and different text sections are represented by rectangular glyphs. The two separate sequences transform into a merged sequence of glyphs which forms a diffline. During the transformation the glyphs of two identical text fragments merge into a single one. The example text was taken and adapted from the beginning of [72].

Different glyph alphabets were designed. Figure 31 depicts three designs that were both promising and unique enough to be tested by users during our pilot phase (see Section 4.6). As a general rule, all diffline designs represent the source document above the suspicious document, following a left/upper=source → right/lower=plagiarism rule, which is consistent with the other views. Thus the top area represents the original document. The lower part represents the suspicious fragment.

(A) RECTANGULAR DIFFLINE Only rectangular shapes of various length and height are used as glyphs. The copy-and-paste sections, depicted as a double height rectangle, are apparent. The rectangles of word sequences that have been modified are shown one above the other in order to make them visually comparable regarding their changes in length. The rectangular version depicts each remaining, removed, or inserted word sequence in its relative length. The accumulated length of the diffline is therefore longer than the representation of either text.

(B) TRAPEZOIDAL DIFFLINE Trapezoidal and triangular glyphs are employed to illustrate the differences in length of modified sections. Triangles represent newly inserted or completely removed text. A glyph that is composed of two triangles shows modifications of similar length. The idea behind this glyph alphabet was to reduce the overall number of visual items. Precisely one item for each kind of event is drawn in order to make the recognition more straightforward.

(C) CONDENSED DIFFLINE This diffline is aligned to the length of the suspicious text and consists of rectangular representations for each section. The darker rectangles show identical text. The light gray rectangles show newly inserted text. The small line-shaped glyphs atop the other rectangles provide hints of textual changes. If the line is as long as the rectangle below, it implies that the original text fragment has been at least as long or even longer than the suspicious one. In favor of using the length of the suspicious document as a reference, we accept slight inaccuracies with respect to the modification operations.

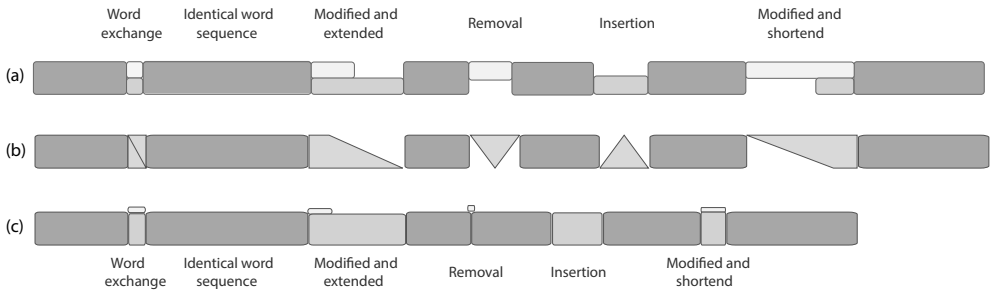


Figure 31: The various difflines composed of different glyph alphabets. All of them represent the same information. The top area represents the original document. The lower part represents the suspicious fragment. (a) RECTANGULAR DIFFLINE, (b) TRAPEZOIDAL DIFFLINE, and(c) CONDENSED DIFFLINE:

The text length of the finding spot is usually encoded as the length of its diffline (see Figure 27(d)). For some tasks however, e.g. in order to facilitate the search and comparison of multiple diffline patterns, it makes more sense to use the entire horizontal space (like in Figure 32(c)). In such cases, we encode the actual length of the finding spot separately as a horizontal bar drawn in the category color below each list entry, whereas the longest finding spot of the document is used for normalization (see Figure 32(b) and also Figure 27(i)).

Our example cases usually contain more finding spots than can be displayed with all relevant information (category, page number, title of possible source, etc) on a regular screen. To see the entire picture and to avoid unnecessary scrolling, we provide means to semantically scale all list entries up or down by hiding or showing less important information, changing font sizes, and adjusting the size of difflines. At the lowest detail level, the diffline is minimized by sliding the upper and the lower part

of a diffline on top of each other so that the copy-and-paste structure remains legible, whereas details about which kind of changes occurred are omitted. The different level of details can also be combined in a Focus and Context view (see Figure 27(j)) where the list entries in the center are given more vertical space to provide additional information while the remaining entries become smaller towards the top and the bottom margin.

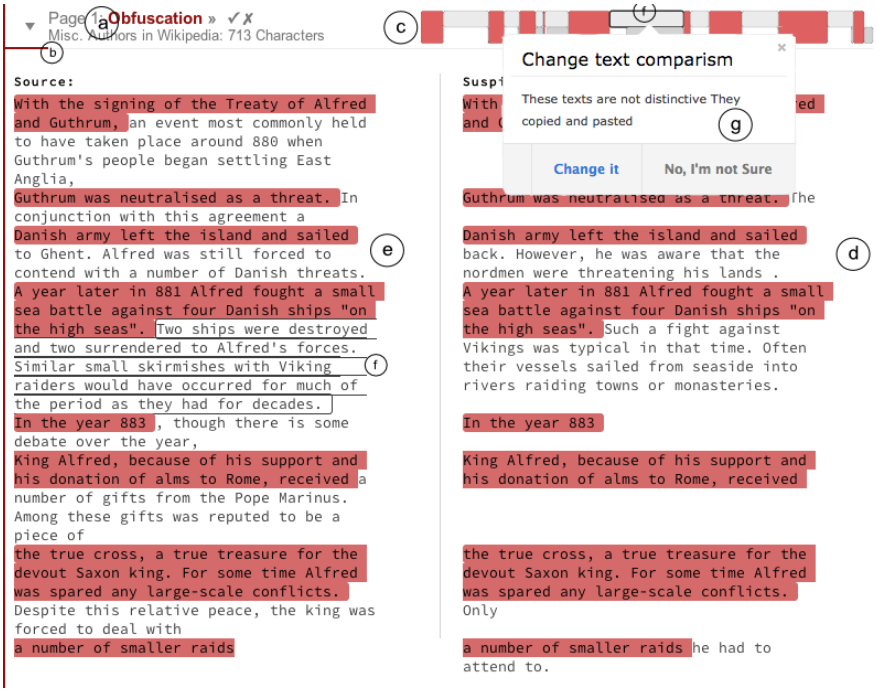


Figure 32: The visual representation of a finding spot shows the essential information on the top left (a) (position and length of the suspicious fragment where length is indicated as a thin horizontal line directly below (b)), plagiarism category (also shown on the left as a vertical line with the color of the category), and the name of the potential original document. The diffline visualization is shown at the top (c). The finding spot is opened and the suspicious text fragment (d), as well as the potential original (e), are shown directly below. The textual view is based upon a particular wrapping intended for easier recognition of the differences and commonalities of both texts. Hovering above a glyph or a text element will highlight both (f) to simplify the mental match.

4.4.3 *The Textual Views*

Although a diffline reveals lots of information about a finding spot and its alleged source, both must be accessible and comparable in a textual form, too. The textual views can be opened on demand and are embedded in the diffline list directly below their respective diffline. We support three different approaches for comparing

text fragments. The first approach resembles the depiction of tracked changes in a word processor (Figure 33). The second approach enables fading in and out the differences between original and suspicious text (Figure 34). Both approaches eventually present variations of a respective text fragment embedded in a single running text, which might be ideal for reading purposes whereas for the diffline approach it seems more promising comparing texts side-by-side with an appropriate wrapping (Figure 32). The wrapping should facilitate the detection of differences and commonalities between texts at a glance and direct the user to the location where reading in detail might be most relevant. In our layout, the identical parts in both texts serve as the skeleton, which is vertically aligned across both texts and highlighted by their background. Modified text blocks are vertically filled so that the corresponding copy-and-paste sections remain aligned. A monospace font with equal character width facilitates judgment regarding how much text has been removed and added or if a substitution of equal length occurred. The color ▬, representing the copy-and-paste sections of the difflines, is used as the background to visually link the structure of the layout and the diffline glyphs (Figure 32(c),(e),(d)). While the copy-and-paste sections, in general, start at the beginning of a line, the modified text blocks in between can also start on a new line or simply at the end of the copy-and-paste section. The latter results in a more compact layout which is useful if the frequency of copy-and-paste sections and modifications is high, e.g., for the plagiarism category Half Sentence Mending.

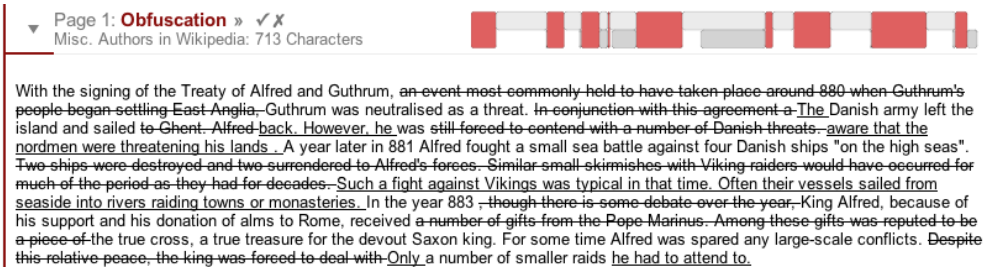
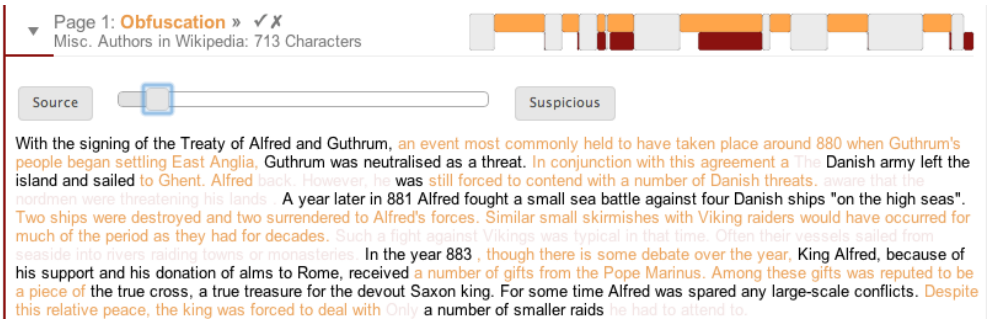
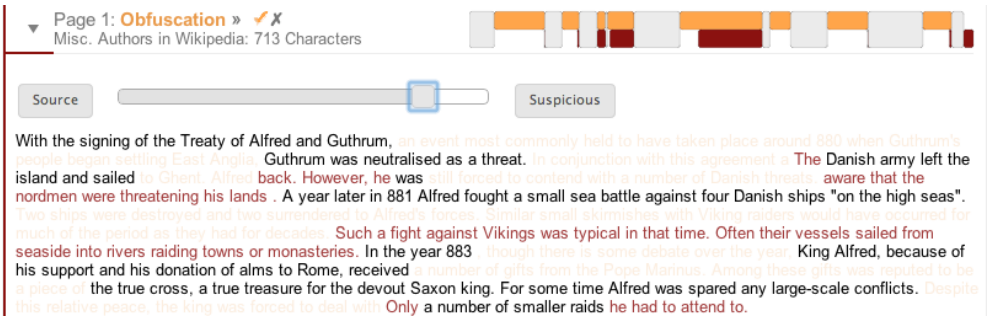


Figure 33: The classic approach for text comparison uses striking out or underlining to reveal removed and inserted words, respectively. The sample text was taken and adapted from [72].

Since our system is intended to support an assessor's workflow, this sometimes means supporting less spectacular and more common interactions which can nevertheless be crucial for improving workflow. Each finding spot shows a set of icons (only at a certain level of detail), such as icons for approving (Figure 27(k) and 34b approved) or rejecting the finding spot, which will then be removed from the list. Another icon enables re-assigning finding spots to other plagiarism types should their current type not be suitable, e.g., for not containing enough copy-and-pasted pieces to be considered complete plagiarism. More importantly, each glyph of the diffline (see Figure 32 (c)), as well as each element in the text view, can be altered (by animated transitions of shape and color). For example, if corresponding text fragments are



(a)



(b)

Figure 34: The diff blending morphs between the two texts by adjusting the transparency values of the respective markups of the changed text. (a) The focus is on the source. (b) The focus is on the suspicious text. Although gaps are created by focusing on the source (a) or on the finding spot (b), the text can be read surprisingly well. Moving the slider from one stop to the other creates an animation-like behavior that draws attention to the changes, especially when using our alternative color scheme.

marked as equal (e.g., by mistake of another member of the community), but instead contain many changes, they can be re-assessed. Conversely, if the the diff algorithm differentiates between text pieces which are, in fact, nearly the same, they can be combined (Figure 32 (g)).

4.4.4 *Color Model*

Although the difflines were designed to reveal the structure of a finding spot even without color, an appropriate color coding facilitates the process of recognizing and interpreting the glyphs. The default color coding displays identical word sequences in difflines and the text views in red ▬ (see Figures 32, 33). The color red emphasizes the fraction and frequency of copy-and-paste actions that were used to assemble a finding spot. It also aids in visually matching copy-and-paste fragments in the diffline with the structured text view. We experienced that pure red looks aggressive and

unpleasant on most displays. Words that appear in only one of the aligned texts were shown in different gray levels (and).

An alternative color scheme (see Figure 35 for comparision and Figure 34 for application) aims to draw attention to the modified parts which might have to be analyzed further. The visual impression is inverse to the first scheme. A neutral gray tone is used for the copy-and-paste passages. A shade of gold-orange is introduced for word sequences that only appear in the alleged original work. It is supposed to express originality and positive character. Text fragments which are only contained in the suspicious work are shown in the category color, which hints at how this modified text segment has been created.

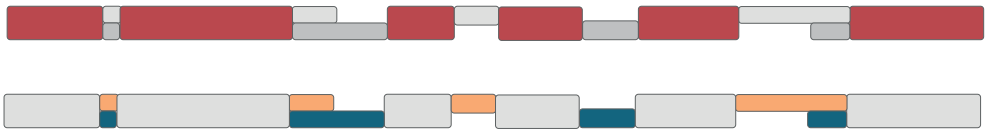


Figure 35: The same diffline with different coloring schemes. The approach at the top focuses on copy-and-paste sections by showing them in red. The lower one reduces the focus on those sections by using a gray value. Attention is instead drawn to the fragments that have been changed by choosing a gold-orange tone for the original text and the color of the plagiarism category (in this case a dark blue-green) for the modified fragment in the finding spot.

We believe that the second color scheme, in combination with the knowledge of plagiarism categories, has some advantages in supporting speculations about changes that might have been made. For example, if the plagiarism category is obfuscation and the text fragment in the finding spot and in the original are of approximately equal length, the finding spot is probably a paraphrased version of the original, which merits closer inspection. We also think that it improves the user's ability to be oriented between the two texts and the diffline (see Figure 32).

We chose to assign colors to the categories (usually less than seven per case) since mapping each source document to an individual color was not appropriate due to the large number of sources in some cases (compare [114]). Colors like the Tableau 20 color scheme [31] were tested but rejected for looking far too positive. Subsequently, the colors were selected by hand to evoke at least a neutral look, or ideally, a negative impression that seems more appropriate in this context. We derived and from for (Almost) Complete Plagiarism and Obfuscation. Pawn Sacrifice and Exacerbated Pawn Sacrifice use different, but familiar, tones to emphasize their commonality, as well as Shake and Paste and Half Sentence Mending . Translation Plagiarism uses a hue which is not related to all others. Although our color scheme narrows down the color space, this was never experienced as an issue, both in our expert reviews and during our lab demonstrations: our color scheme maintains a reasonable level of discrimination.

4.5 DATA PREPROCESSING AND IMPLEMENTATION DETAILS

The alleged cases of plagiarism are publicly available at the aforementioned wikis [70] and [71]. We acquired the underlying data via the Wikia-API. Since the cases have been entirely manually annotated with very limited templating support from Wikia's Wiki software, many inconsistencies with regard to naming schemes, tags, typos, encodings, etc. remain. All of these issues cause little disruption to the Wikis since the Wiki software handles them gracefully, but they foreclosed our attempts to process the raw data automatically. We therefore systematically reviewed the plagiarism cases and semi-automatically removed inconsistencies by hand, sometimes using Python scripts. As a result of roughly 180 hours of student work, a total of 41 plagiarism cases containing nearly 6100 finding spots (with an average of 6200 words per spot) that link to over 950 sources are now available in a consistent JSON format.

Our prototype is entirely web-based and both its logic and presentation layer are executed at client side, whereas the server only delivers the web page along with the required script files. The JSON files of the finding spots are dynamically prefetched during scrolling and filtering operations before the respective difflines come into view. The system has been developed and tested with recent versions of the Chrome web browser. Four JavaScript libraries were used: jQuery for accessing the DOM-Elements more conveniently, low-level methods of D3 for structuring and wrapping the drawing operations, the google-diff-match-patch library to determine text changes between a finding spot and its original, and Backbone.js for MVC support.

4.6 DIFFLINE DESIGN DECISIONS

Several glyph alphabets were designed to express what changes might have occurred between two texts. Three designs that were most promising were chosen based on their respective features (see Figure 31): (1) the rectangular difflines because of their simplicity, (2) the trapezoidal diffline due to their seemingly expressive glyph alphabet, and (3) the condensed difflines because of their compactness. A pilot study was conducted to obtain feedback about their general usability, their comprehensibility, and which of them should serve as default. We chose a between-group design with 18 participants. Our rationale for doing so was due to the fact that being briefed in two or more diffline alphabets causes confusion: similar visual elements were used across the alphabets, and a strong learning effect occured from performing the same task consecutively, albeit with different alphabets.

Each participant accomplished three different tasks. Prior to these tasks, the participants were thoroughly briefed about the characteristics of the particular diffline used in his/her group by exploring two different example difflines along with their corresponding finding spots consisting of the source and suspicious text fragments. For the

first task, the glyphs of four difflines had to be assigned to their matching text pieces. These difflines varied in word length and structure (approximately 14-21 glyphs per diffline, a representative number). As for the second assignment, the participants were supposed to visually examine another four difflines—glyph by glyph and without accompanying text—and to explain what changes could have possibly happened in terms of expressing the changes that occurred between two text fragments. Finally, the participants answered a questionnaire about how difficult they found the assignments, how useful they found the glyph alphabets, and what general improvements they propose.

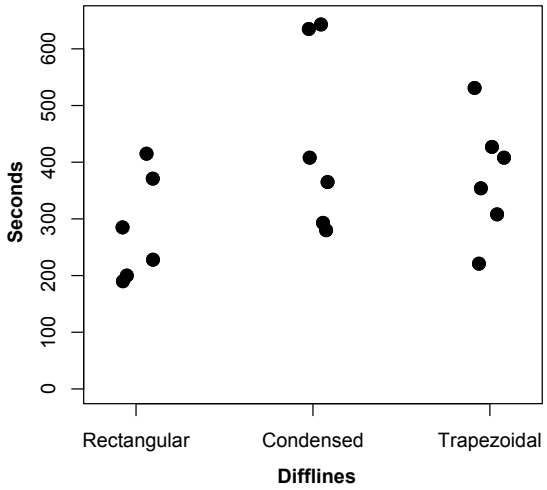


Figure 36: Task completion times for assigning the text fragments and glyphs (task one).

The difficulty of task one (assigning text fragments to glyphs) was assessed by the participants with means of 1.6, 2.0 and 2.6, respectively, for rectangular, trapezoidal, and condensed difflines. This coincides with the measured task completion times depicted in Figure 36. Their means of 281s, 387s, 437s reveal a similar pattern, although it did not become statistically significant in an ANOVA which is likely due to the small group size and the intra-group design. We also observed how often the participants corrected their assignment during the task. The average values of 2, 2.6 and 3 self-corrections fit in that order, too. Nevertheless, a very positive result was that we recognized only 2 errors overall in the completed first tasks.

Some difficulties occurred while interpreting and orienting the short rectangles for the condensed difflines and comprehending the meaning of the orientation of the triangles in the trapezoidal ones in task two. Although, it was rated by all groups as being similarly difficult with an identical mean of 2.0 for each of the three groups the error rates for task two showed a different picture. Overall, 71 glyphs had to be assigned, 36 of which did not represent identical text fragments. The rectangular version was most easily understood and resulted in no errors when interpreting glyphs, while an

average of 2 and 4.3 errors were made for the trapezoidal and condensed difflines, respectively.

Overall, the answers to the question of how useful and comprehensible the glyphs of the respective groups were show that the rectangular difflines were appreciated most (mean of 1.3), whereas the other approaches were judged as being less comprehensible (mean of 2.3), which follows prior results.

Although nearly half of the participants recommended the usage of colors as a very helpful improvement, the results of rectangular difflines show that gray levels (no colors at all) are sufficient for the specific tasks of this study. Altogether, we chose the rectangular version as default one and used color to highlight copy-and-paste fragments in a finding spot.

4.7 EXPERT REVIEWS, FEEDBACK AND FINDINGS

After the main functionality of the system was developed (overview, difflines, textual views, basic interaction), we reviewed the system with three external experts. One writes plagiarism assessments for a living and the others are very active in the German anti-plagiarism community. Even though they have been very active in the community for many years, they have only used static visualization thus far. Therefore, they enjoyed the general interactivity of the system and its different views of a case. Every one of the experts immediately tried to locate particular finding spots they were familiar with. In this regard, filtering and exploring by sources seems what interests them most, especially identifying and filtering by the finding spots of these sources that were used most in the suspicious document. Their favorite features are:

- Having the ability to see all finding spots at once
- Being able to trace the finding spots back to their sources without, for example, recalling a particular color coding (like in [114]).
- Being able to recognize relationships between the distribution across the entire document and particular categories (e.g. Figure 37).
- Having easy access to small sets or individual finding spots via fluent interactions and filtering capabilities is a clear advancement over the existing visualizations.

They were particularly fond of the diffline idea, which they found clear and legible for the intended task of providing a visual pre-assessment to decide which spots are more ambiguous and should be further investigated in detail with the help of the textual view. Two of the experts liked the special wrapping of the text view using the equal parts as a skeleton, whereas the third was more fond of the text blending method.

Another experience during the reviews was that, after becoming more familiar with the prototype, the experts started exploring and comparing different cases and discussing their peculiarities (Figure 37 contrasts cases with different properties). They further suggested introducing an ordering in decreasing length of finding spots grouped by most used sources in order to speed up the review process: if larger fragments are confirmed to be plagiarism, smaller ones can be postponed. In this regard, they preferred an absolute encoding of the length of a finding spot and recommended introducing a possibility to adjust a length threshold to filter finding spots that are too short to be of use.

During lab tours, our prototype became one of the most discussed exhibits. All of our guests were very interested in the matter regardless of profession or field of research. Our guests are usually surprised by the severity of some of the cases (which is made apparent by the overviews) and the pettiness of others, whereas both have received comparable media attention. Often visitors explored details with the difflines and textual views on their own after being shown an example. Actually, some of them started comparing different cases regarding the distribution and number of finding spots and original sources, a task our prototype was not originally intended for.

For example, Figure 37 shows a comparison of different methods of plagiarism, where each case differs in length, number of finding spots and sources, categories, and distribution of finding spots. (a) The very few crossings indicate that this suspect worked in a linear way, integrating source by source after another. (b) Only few sources suffice if they can be exploited extensively. (c) This short document employed surprisingly many sources. (d) The suspicious document utilizes a main source across all pages (selected), which indicates that the overall structure of the original work was employed and filled in with other sources (not selected).

Furthermore, Figure 29 shows such a comparison of two examples (a-c) and (d-f) and reveals insight into the structure of the suspicious documents. Although in both cases the amount of finding spots is high and evenly distributed along the pages (a and d), the upper visualization shows that, in almost every page, a finding spot occurs whereas in (d) it only appears in every third or fourth page. In both cases, the length of the finding spots are similarly distributed. The color impressions of both cases already indicate that different types of plagiarism have been used. Subfigures (c) and (f) confirm that. The upper case shows that most findings spots were categorized as obfuscation or as complete plagiarism without referencing their originals. In the lower one, most spots are pawn sacrifices and, therefore, the references the texts have been taken from are contained in the bibliography, only referenced at other positions in the document. For only a few pawn sacrifices, such finding spots can be considered accidental and unintentional. However, for such a substantial amount, this is most likely not the case.

We originally expected that the difflines reveal distinctly different patterns between categories, e.g. more frequent text modifications in the category Obfuscation or that

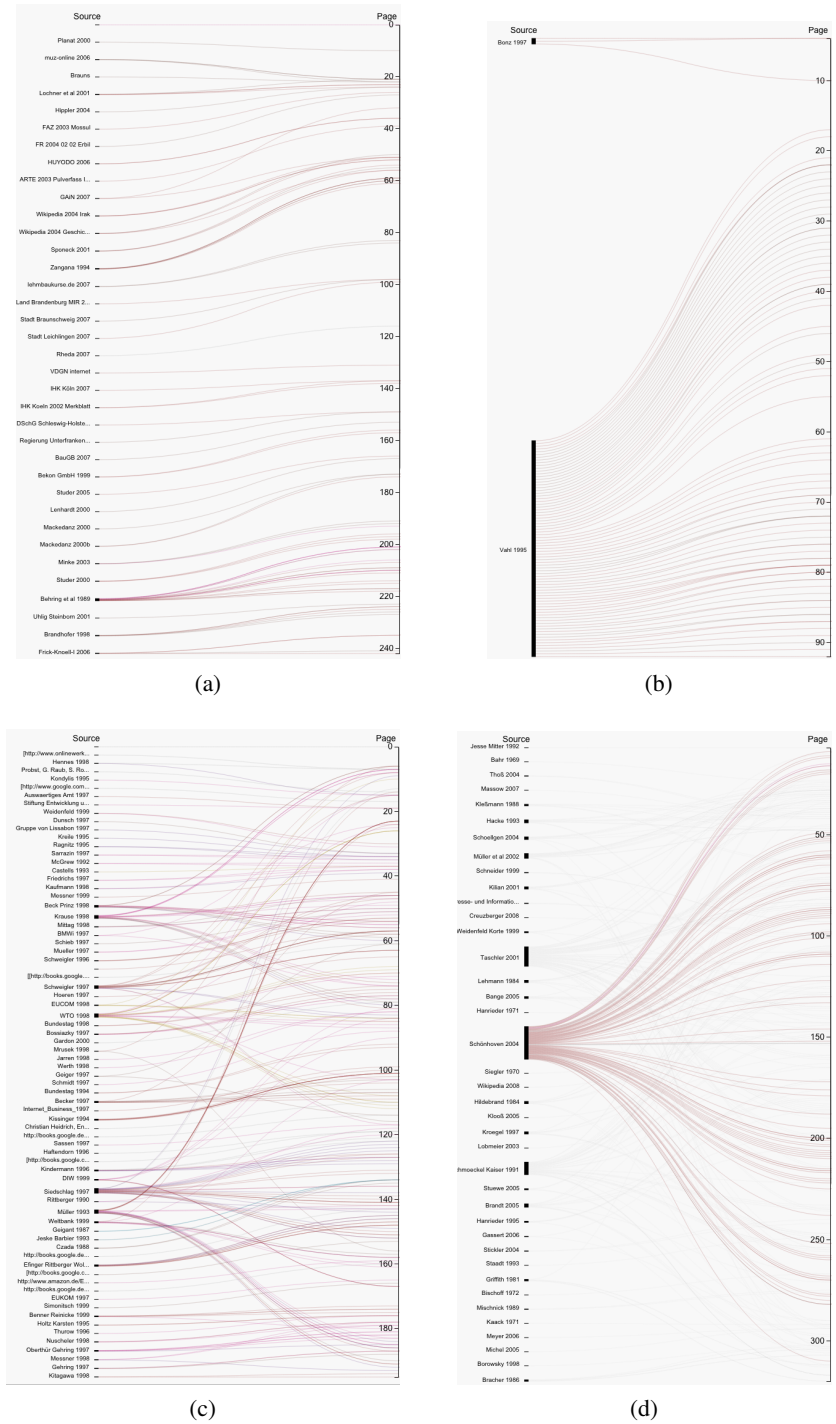


Figure 37: Different methods of plagiarism. Each case differs in length, number of finding spots and sources, categories, and distribution of finding spots: (a) Linear Manner, (b) Only few sources, (c) Many sources in a short document, and (d) A main source across all pages serves a content structure

the difflines look quite similar with only very few modified text fragments for the category Complete Plagiarism. In some cases, one can see quite consistent patterns but, unfortunately, quite often the categories show a wide spectrum of copy-and-paste patterns which leads to interesting questions, such as: are the categories defined by the community itself not discriminative enough? Are they too fuzzy in description, or was the community sloppy in ensuring a consistent categorization?

Another finding is related to the data preparation done by the students. The textual visualizations especially the side-by-side view served as a very useful tool to verify how thoroughly they did the preparation. If the markup did not match the text, the data was not properly converted.

4.8 CONCLUSIONS AND FUTURE WORK

We present a new approach for the interactive visual analysis of alleged cases of plagiarism. Our interface is based on three levels of abstraction. Our overview displays provide information regarding the structure of the document, the specifics of the finding spots, and how they are related to the original works and authors. The list of difflines provides a compact overview of finding spots, reveals plagiarism patterns by visually encoding the differences and similarities of two text fragments, and directs attention toward further analysis. To this end, a textual side-by-side comparison of original and finding spot can be shown to enable their direct comparison. Our prototype provides effective means to navigate and filter the finding spots and enables direct interaction between finding spot, original, and their diffline. As our study shows, users became quickly proficient with our system and were able to correctly interpret difflines. Furthermore, the reviews by our plagiarism experts confirm that our tool is far more effective than existing static and non-static visualizations. Therefore, we believe that a visual analysis tool like ours will play an important role to verify plagiarism allegations in an effective manner and to convincingly present the evidence to councils or even to the general public.

Further development of natural language processing technologies will possibly lead to automatic categorization of finding spots which is potentially more precise than the community members are today. We are also working on detection algorithms for obfuscation techniques employed in paraphrased text, such as utilizing slightly different words with the same stem, converting verbs into nouns and vice versa, or using synonyms. The difflines should be extended to express and reveal passages that were created with such modifications. However, a particular challenge is the uncertainty that comes with a machine-generated categorization. Another aspect that should be addressed in the future is the visualization of nonlinear paraphrasing where particular word sequences are shuffled or rearranged in order to mimic autonomous reasoning and deducing (see Figure 38). Although barely existing in our manually categorized cases (even in the Shake and Paste and Half Sentence Mending categories), we are

certain that such less obvious approaches are used in more cleverly plagiarized documents.

Figure 38: Proposal for nonlinaer difflines to reveal nonlinear mixing of text fragments from a source document in a finding spot.

Although our current tool contains some capabilities for group reviews, such as approving finding spots or changing their categorization, other operations to manage complex alleged plagiarism cases are needed: foremost proper user management, as well as an additional top level view, in which several suspicious cases can be depicted at once in an effort to compare them regarding their topics, methods of plagiarism, or shared sources. Once such capabilities are available, further tests involving the anti-plagiarism community and an integration with our Picapica software are intended.

Part 5

CONCLUSION, DISCUSSION, AND FUTURE WORK

5.1 MATCHING DECISION THEORY

At the beginning we introduced the different models of decision-making by Dewey, Simon and Brim. Regardless of order, number, and concurrency of the steps of these models, one can infer important subtasks that are supported by the visualizations of this thesis (see Table 4). Due to the fact that, for every visualization the problem area or the challenge is almost entirely clear, it is not necessary to identify the problem itself and, therefore, there is no need to support the first step in the mentioned models. The Wordgraph is intended to support people that are uncertain about the customariness of certain phrases. The Product Explorer facilitates a given problem, too, deciding among numerous product alternatives on the market. The plagiarism visualization also tries to guide the user faced with making an actual decision. Did the author of this work plagiarize or not?

Dewey	Simon	Brim
(1) Feeling a difficulty	(1) Intelligence	(1) Identification
(2) Defining the character		(2) Obtaining necessary information
(3) Suggesting possible solutions	(2) Design	(3) Production of possible solutions
(4) Evaluating suggestion		(4) Evaluation of such solutions
(5) Further observation, acceptance or rejection	(3) Choice	(5) Selection of a strategy

Table 4: Overview about what steps of the different models of decision-making being supported by the ○ Wordgraph, the ● PRODUCT EXPLORER and, the ● PLAGIARISM VISUALIZATION.

The next steps are more complicated in assignment. One can argue that Dewey's second step, defining the character of the problem, is somehow partially covered by the Wordgraph due to its ability to process queries wherein users have expressed their uncertainties using different wildcards. In this regard, the Wordgraph also supports Simon's step of designing a possible solution, but also considers building up several alternatives by looking at the resulting graph-like visualization, which is defined as an extra step in Dewey's model.

A similar reasoning can be applied to the Product Explorer. The user is also able to express his wishes regarding the attributes of a product based on a visual query interface. Users can quickly narrow down the product search to a small subset, suggested visually by greying out all paths that do not match those requirements, which then involves the third step of Dewey's model, as well the second one of Simon's. A different perspective is also possible since the Product Explorer, as a whole, suggests

potential solutions. Additionally, using the visual query interface can be considered as evaluation step (Dewey) and choice phase (Simon).

The Plagiarism Visualization is unambiguous in this regard. Albeit different reviewers or community members might have different options, eventually, there are only two possible outcomes, either he/she re-used text from other work(s) without properly mentioning and referencing those texts or he/she did not.

As one can see in Table 4, Brim seems to be the most fitting model to all systems. All systems can provide the necessary information either as indexed corpus of phrases, as dataset of products, or as a collection of finding spots. The visualizations present the possible solution in a digestible manner, whereas the interactive features (navigating, exploring, and filtering) support the users in evaluating the solutions and selecting an appropriate alternative.

In the introduction, I argued regarding to the cyclic proposal by Mintzberg, Raisinghani, and Théorêt [69] that loops within the models are not a real issue, when they are covered by appropriate interaction that supports such non-linear scenarios. First and foremost, the Product Explorer is particularly suited for a cyclic decision process. At any time, the user is able to not just revoke the most recent attribute constraints (slider or extended areas) that were picked right before, but to enable and disable any attribute with an immediate visual response. The Wordgraph provides a similar interaction scheme since exploring possible subgraphs down to possible solutions of the result set is reversible even up to rephrasing the query.

On the contrary, assessing alleged plagiarism should be a linear process: ideally, after thoroughly scrutinizing and assessing each single finding spot, the final verdict is made. Our system supported this scenario, initially. Unfortunately, it does not work this way in reality, as our expert reviews revealed. Often due to the lack of time, the assessors concentrate on the biggest fragments at the beginning and those that are most clearly believed to be an immoral re-use of text. This approach follows a simple, yet convincing and time-saving, strategy. Once a certain fraction of the document is proven as having been plagiarized, the entire document can be considered plagiarized, regardless of how many smaller finding spots may exist and have not been assessed. Therefore, the reviewer jumps back and forth within the document searching and assessing only particular finding spots. This kind of exploration is also supported by being able to order and to filter according to categories, page ranges, original works, as well as the size of the finding spot. Furthermore, the diffline representation helps the user to decide which spots must be investigated in detail with the help of the text view and which spots are assessable only by the glyph representation, another time-saving factor.

Applying the knowledge of Knight's [56] conditions to our prototypes, we can infer the following picture: the Product Explorer can be assigned mostly to the certainty condition. We can see, or at least estimate, the consequences of the chosen product

since they are described as its attribute manifestation (if the product specification of the manufacturer is trustworthy). For example, choosing a lightweight notebook (in our default data set) usually means being restricted with regard to performance critical attributes like CPU type or a dedicated graphic card, but also with display size or battery time. Of course, some aspects remain uncertain (and not even their risks can be estimated without doubt), such as assembly conditions and quality of the components.

The alternatives of assessing a case of possible text re-use are as evident as their respective consequences are, as well. Thus, it belongs to the certainty condition. The Wordgraph, however, cannot be assigned to any of the conditions easily since non-native speakers often do not have the knowledge or the innate sense of language as native speakers do. They are not even aware which alternatives they have in wording and, therefore, none of the conditions can really be applied. This issue occurs before Knight's model starts since the users are not informed at all about the alternatives. Considering this, the main task of the Wordgraph is to inform, which means to acquire and to present information, and to proceed with or even start a decision process. One can argue similarly for the Plagiarism Visualization. The alternatives are obvious and severe: hence, one has to focus on having and keeping the reviewer informed throughout the assessment about the entire picture, as well as the details of the case. Although the Product Explorer matches Knight's model best, it is also intended to inform the user about the situation of a certain market segment to choose from. Visually emphasizing the consequences while choosing alternative results from the multivariate character of the data is, nevertheless, a very convenient side effect.

5.2 EVALUATIONS

All three prototypes were evaluated with useful means. User studies were conducted to gain insight into the performance and usability of the Wordgraph and the Product Explorer against established interfaces. This meant, on one hand, acquiring the user's interface preferences between the Netspeak web interface and the Wordgraph and, on the other hand, measuring task completion times between the Product Explorer and an advanced web shop interface at that time (*notebooksbilliger.de*, 2011).

The Product Explorer significantly outperformed its competitor concerning both task completion times and user preference. Admittedly many web shop interfaces have significantly improved, especially, with immediately responding graphical elements or even with scented widgets [130]. Some badly designed shops still exist, though. Conducting a similar experiment again might yield comparable task completion times. However, I believe that advantages with respect to overview and orientation will remain for the Product Explorer when narrowing down the initial set: for example, when disabling prior choices or when anticipating which further choices are possi-

ble. A long result list that has to be scrolled through cannot offer such insights. This should not be underestimated regarding the completion time.

Evaluating the Wordgraph by measuring task completion times would not suffice. Time is not the issue here. The question is how well the users are feeling informed by the interfaces when the potential complexity of result sets is considered. The test setup aimed at that issue. The preference for the Wordgraph interface increased along with the number of used wildcards and was significantly higher for two and three wildcards than was the web interface. This coincides with the features appreciated most by the participants, such as having: (1) an overview with the most important information to start from and (2) the possibility of exploring the response set fluently in detail by applying subgraph filtering. Interestingly, the participants followed the Mantra of Shneiderman [99] "Overview first, zoom and filter, then details-on-demand" without even realizing it.

Due to the specific domain, as well as the lack of a comparable system, the plagiarism visualization had to be reviewed as a whole, by experts. The difflines, however, could be tested by regular people since the glyph alphabet is a more general concept and not limited to experts. Although the results between the alphabets were not significant in an ANOVA, the overall outcome, including how well the participants understood the task, the low frequency of errors they made, and the pace in which the participants proceeded during the tests, showed us being on the right track in developing a visual abstraction of the textual representation of the finding spots.

The experts clearly stated that the advantages of the prototype, compared to all other (mostly static) solutions, included general interactivity such as the filtering and exploring capabilities starting from all finding spots at once down to small sets or individual finding spots. Additionally, it was beneficial to be able to recognize distributions of finding spots across the entire document or to trace the finding spots back to their sources. They also found the difflines concept of providing a visual preassessment of the finding spot, as well as the special text wrapping, clear and legible and thus very convincing.

5.3 VISUAL PRINCIPLES

One main element all visualizations have in common is heavily relying on establishing visual links for depicting the most important relations of the datasets, such as occurrences of sources, co-occurrences of words, and concatenations of attributes. Particularly, all three systems are layered structures either in horizontal or vertical orientations. Even more, one could consider them as layered graph-like structures when thinking of the extended areas of the Product Explorer as nodes/vertices with an equal number of incoming and outgoing edges (see also Section 3.7) or when

thinking of the entire diffline list as special kind of layer connecting the currently-shown elements via edges to the overview. All connections are drawn as cubic Bezier curves with a similar pattern of positioning the control points, which may reflect aesthetic affectations of the author, however, this kind of drawing was also preferred by the participants of the Product Explorer evaluation.

Another principle of all systems is the attempt at avoiding additional views, particularly since, in my opinion, well-partitioned and connected structures in a single view gain advantage over different ones linked by visually highlighting the elements within many views. There are only two exceptions to this rule that are both meaningful and well-reasoned, which can be seen in Figure 22 and 23 on page 56: (1) The exclusive decision bar below the main view of the Product Explorer is meant as conceptual barrier between the current attribute axes shown in the main view the user is focusing on and the attribute decisions the user was so certain about that the axes could be moved from the main view into the exclusive bar. Similarly, the axis repository is intended to contain all attributes of lesser importance, so they could be dragged into the main view on demand. Eventually, a single axis can only appear in one of the three views, but not in more, which is why no additional linking across the views is necessary. (2) The category bar of plagiarism visualization replaces the categories on the right hand side where they are connected to the currently displayed difflines if the horizontal space does not suffice. Due to the low number of categories and to avoid introducing additional clutter by drawing edges orthogonally to the existing ones, such a trade-off seems reasonable.

5.4 CONTRIBUTIONS

This thesis contributes three interactive visualization systems carefully designed to cope with actual decision-making problems concerning important issues such as multivariate choice, overcoming uncertainties, and consideration and assessment.

The Wordgraph, a dynamic graph visualization for interactive exploration of search results for complex keywords-in-context queries, has been presented to support non-native speakers in finding customary phrases. The contributions, in detail, are: (1) a novel method of transforming a list of phrases into a legible column-based graph representation where multiple occurrences of words are aggregated into scalable visual word items that are sorted per column, (2) layouts for arranging the columns and words in combination with vertical orderings, (3) different views for edge drawing, as well as a method for minimizing edge crossings, (4) appropriate techniques to filter, navigate, and expand the graph, (5) a study comparing the Wordgraph visualization and the textual Web interface (both based on the Netspeak service) that showed the advantages of the Wordgraph with an increasing number of wildcards, and, (6) additionally, typical retrieval tasks, as well as user scenarios, that have been revealed by analyzing the query logs of the Netspeak service.

The presented Product Explorer is an interactive visualization for choosing a certain product over numerous alternatives sharing the same attribute set. This has been accomplished by: (1) extending Parallel Coordinates with so-called extended areas to depict categorical and ordered data with only few occurring values in a meaningful manner, (2) focusing on a single task that guides the user with a visual query interface while expressing his/her requirements regarding the desired attribute variations and, therefore, ultimately narrowing them down to a fitting subset or even one remaining product, and, (3) introducing an exclusive decision technique that is able to dismiss a particular axis that is no longer needed if the user is very certain about the chosen value of that attribute. Therefore, in combination with the (4) visual repository (that contains axes of lesser importance), the number of attributes and products to be considered will be reduced. (5) Subsequently, the performed user study confirmed that the Product Explorer is indeed an excellent tool for its intended purpose for casual users. The study also reveals that the users prefer cubic curves in combination with extended areas because it led to a tidier display of a large number of products by reducing occlusions near an axis.

The central contributions of our plagiarism analysis tool include (1) a new three-tiered approach for depicting and exploring alleged cases of plagiarism (2) acquired and converted from the most elaborate collections of suspicious PhD theses on GuttenPlag [70] and VroniPlag [71], (3) an overview paradigm for navigating and selecting subsets in a suspicious document by category, page range, or source document, (4) diff-flines as an effective glyph-based abstraction of differences and similarities between the suspicious text and the alleged original, and (5) the support for fluid and coherent interaction between the different levels of detail. (6) The reviews with our plagiarism experts confirm that our tool provides a significant improvement over existing static visualizations in that regard, as well as its capability to improve their workflow for forming an opinion about a plagiarism case. This is especially true regarding time savings during that process, particularly if a whole council or committee has to come to a verdict.

Furthermore, the three decision problems were framed in the context of established decision-making literature by discussing them according to the steps of the most common models, considering them in light of one of the most important non-sequential approaches, and taking a closer look at the conditions of the particular decisions. Subsequently, relevant aspects of the evaluation were reflected, the visual principles shared by all visualizations concluded, and, lastly, future opportunities will be considered.

5.5 THE SHAPE OF THINGS TO COME

As with most works, an open field of future opportunities remains, such as improving specific aspects of, or integrating novel features into, the presented systems, some

of which have already been discussed to a certain extent within particular parts of the thesis. However, within a wider perspective, one should also address additional opportunities.

One possibility is a survey based on real-world decision processes that guide the user toward a fitting visualization paradigm or interface. Maybe an internet page comparable to existing sites presenting time-based visualizations [111] or textual data [37], along with an appropriate filtering interface, might be worth the effort.

Introducing textual and recommendation knowledge to improve the Product Explorer is an obvious and welcome next step. This information can either be integrated as additional (maybe specialized) axes or as interactive coordinated multi-views. Categorical or rating information that is usually used to assess a product's quality (most often one to five stars) can be crawled, normalized, and integrated as axes into the Parallel Coordinates display. Even some aspects of the actual reviews, such as their length, might be usefully depicted as axes. However, for analyzing the sentiment of all reviews of a certain product or for searching based on the Netspeak technology within these reviews, a single axis could not express such information. Therefore, novel views have to be added.

The interface of the plagiarism visualization would also benefit from enhanced search capabilities beginning with simple search and filter and advancing to more complex operations such as marking dubious word sequences and finding phrases with similar patterns (for example, by using wildcard-based Netspeak technology) in the suspicious document, as well as in the sources.

Another possible research direction can be seen in considering the extended Parallel Coordinates display of the Product Explorer as a layered graph, which results in a more legible representation of all products by computing a proper positioning of the crossing points or extended areas for certain types of axes, such as categorical or even ordered ones. First attempts have already been made (see Section 3.7), but lack an evaluation with appropriate measures. Unfortunately, the Pargnostics screen-space metrics proposed by Dasgupta [18] for Parallel Coordinates did not fit in our case. Instead, I propose an alternative based on the idea that tracing a single line, as well as lines with similar paths, is less difficult if the vertical offsets between the values of succeeding axes are minimal and the resulting line is less oscillating. Therefore, a target function should aim at minimizing all vertical offsets of all depicted product paths. The approaches described in this thesis must therefore be tested with different plots containing various assortments of the axes itself, as well as different assortments along the individual axes against the target function.

Additionally, it may be with the effort to utilize some kind of semi-automatic layout including features that enable the users to rearrange the extended areas arbitrarily along a particular axis by themselves. The ordering of the connected extended areas at adjacent axes would follow according to the number of connected paths or other

criteria. With such capabilities, every user is able to customize his or her own search session by reordering the plot. This personalized plot could then serve as a default for a subsequent session.

Furthermore, some ideas exist for merging features of Parallel Sets [59] or of interactive Sankey Diagrams [90] into the Product Explorer in order to aid users during the their first steps of the selection process. Depicting ribbons instead of numerous single paths, initially, reduces the visual complexity of the overview. While choosing the first attributes, these ribbons should remain and must only decrease in width (or parts of the ribbon must be greyed out) to reflect the narrowed-down product set. The crucial points seem to be determining after how many steps a transition to a path's view is advisable and how such a transition should ideally appear. A web-based reimplementation could benefit from possible performance advantages against single path drawing.

So far temporal aspects have not been addressed at all, which was simply due to the lack of time-varying aspects in the given data. Visualizing time-varying data with Parallel Coordinates is still a hot topic, particularly with respect to exploring simulation data sets. Based upon the visual ideas of Product Explorer concerning path drawing and axis depiction, a coordinated multi-view framework is being developed within our group that aims at massive time-varying data sets.

As for the Wordgraph interface, taking time into consideration is only possible with a specialized corpus (or an integrated index built upon several corpora) that provide(s) data for certain ages in history of the same language. A tool for exploring differences in customariness of phrases and wording between those ages might be interesting for linguists and for historians, too.

The difflines, in their current form, can only display two discrete states – the original text versus the alleged text reuse. Extending the diffline concept toward showing multiple revisions at once in order to unveil how a text has been rewritten is surely an interesting challenge. An appropriate corpus aimed at text reuse was already gathered [84] by our colleagues of the Web Technology and Information Systems group at the Bauhaus-Universität Weimar. The corpus consists of manually written revisions during a special kind of task where writers (hired at the crowdsourcing platform *oDesk.com*) had to search for possible sources from a third party for a given topic and, subsequently, had to compile a new document reusing and paraphrasing the texts they found. On the one hand, animated transitions could be employed for visualizing the changes between successive or even arbitrary revisions. However, focusing on several moving glyphs at once might be an arduous task, which is why a special kind of sequence throughout the transitions has to be applied. A more static approach, such as a diffline stack, might be an alternative. It might also be an advantage over the well-known HistoryFlow [119] since, in my opinion, it provides a more legible alignment of the glyphs instead of adjusting all glyphs to one side only (left or top) without any space between the single glyphs.

Of course, some of the ideas are quite vague still, including merging features of the Wordgraph and the Plagiarism Visualization; some others are more clear and have been scheduled, such as the further development of the difflines or the successor of the Product Explorer aimed at simulation data (which is not far from being submitted and hopefully published). Nevertheless, the intelligent support of decision processes through appropriate visual metaphors and visual analytics tools is is still largely unexplored and offers many challenges and opportunities for further research.

BIBLIOGRAPHY

[1] David F Andrews. Plots of high-dimensional data. *Biometrics*, pages 125–136, 1972.

[2] N. Andrienko and G. Andrienko. Informed spatial decisions through coordinated views. *Information Visualization*, 2(4):270–285, December 2003.

[3] Jeanette Bautista and Giuseppe Carenini. An integrated task-based framework for the design and evaluation of visualizations to support preferential choice. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '06, pages 217–224. ACM, 2006. ISBN 1-59593-353-0.

[4] D. Belazzougui, F.C. Botelho, and M. Dietzfelbinger. Hash, displace, and compress. In *ESA '09: Proceedings of the 17th European Symposium on Algorithms*, pages 682–693, Springer Berlin / Heidelberg, 2009. Springer. ISBN 978-3-642-04127-3.

[5] Enrico Bertini, Andrada Tatu, and Daniel Keim. Quality metrics in high-dimensional data visualization: an overview and systematization. *IEEE TRANS. ON VISUALIZATION AND COMPUTER GRAPHICS*, 2011.

[6] Thorsten Brants and Alex Franz. Web 1T 5-gram Version 1. Linguistic Data Consortium LDC2006T13, Philadelphia, 2006.

[7] M. Brehmer, S. Ingram, J. Stray, and T. Munzner. Overview: The design, adoption, and analysis of a visual document mining tool for investigative journalists. *IEEE Trans. Visualization and Computer Graphics (TVCG / Proc. InfoVis)*, 20 (12):2271–2280, 2014.

[8] Orville Gilbert Brim. *Personality and decision processes : studies in the social psychology of thinking / [by] Orville G. Brim, Jr. [and others]*. Stanford University Press Stanford, Calif, 1962.

[9] Michael J. Cafarella and Oren Etzioni. A search engine for natural language applications. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 442–452, New York, NY, USA, 2005. ACM. ISBN 1-59593-046-9.

[10] Giuseppe Carenini and John Loyd. Valuecharts: Analyzing linear models expressing preferences and evaluations. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '04, pages 150–157, New York, NY, USA, 2004. ACM. ISBN 1-58113-867-9.

[11] Linda Case. *Dog food logic : making smart decisions for your dog in an age of too many choices*. Dogwise Publishing, Wenatchee, Washington, U.S.A, 2014. ISBN 1617811386.

[12] Remco Chang, Alvin Lee, Mohammad Ghoniem, Robert Kosara, William Ribarsky, Jing Yang, Evan A. Suma, Caroline Ziemkiewicz, Daniel A. Kern, and Agus Sudjianto. Scalable and interactive visual analysis of financial wire transactions for fraud detection. *Information Visualization*, 7(1):63–76, 2008.

[13] Herman Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68(342):361–368, 1973.

[14] Fanny Chevalier, Pierre Dragicevic, Anastasia Bezerianos, and Jean-Daniel Fekete. Using text animated transitions to support navigation in document histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 683–692, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9.

[15] Pirooz Chubak and Davood Rafiei. Index structures for efficiently searching natural language text. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 689–698. ACM, 2010. ISBN 978-1-4503-0099-5.

[16] Christopher Collins, M. Sheelagh T. Carpendale, and Gerald Penn. Visualization of uncertainty in lattices to support decision-making. In *EuroVis*, pages 51–58, 2007.

[17] Cristina Conati, Giuseppe Carenini, Enamul Hoque, Ben Steichen, and Dereck Toker. Evaluating the impact of user characteristics and different layouts on an interactive visualization for decision making. *Comput. Graph. Forum*, 33 (3):371–380, 2014.

[18] Aritra Dasgupta and Robert Kosara. Pargnostics: Screen-space metrics for parallel coordinates. *IEEE Trans. Vis. Comput. Graph.*, 16(6):1017–1026, 2010.

[19] John Dewey. *How we think*. D.C. Heath and Co Boston, 1910. URL https://archive.org/details/howwethink000838mbp.

[20] D Dorling. Cartograms for human geography. *Visualization in geographical information systems*, pages 85–102, 1994. URL http://books.google.de/books?id=z_tOAAAAMAAJ.

[21] David Dun. *Choosing kitty: Making decisions about kitty caring*. CreateSpace, Scotts Valley, CA 95066, U.S.A, 2015. ISBN 1505897521.

[22] Ward Edwards. The theory of decision making. *Psychological bulletin*, 51(4): 380, 1954.

[23] Geoffrey Ellis and Alan Dix. Enabling automatic clutter reduction in parallel coordinate plots. *IEEE TVCG*, 12:717–724, September 2006. ISSN 1077-2626.

[24] Niklas Elmqvist, Pierre Dragicevic, and Jean-Daniel Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1539–1148, 2008.

[25] John W Emerson. gpairs: The generalized pairs plot. http://cran.r-project.org/web/packages/gpairs/gpairs.pdf, 2015. [Online; accessed 2015-04-27].

[26] John W Emerson, Walton A Green, Barret Schloerke, Jason Crowley, Dianne Cook, Heike Hofmann, and Hadley Wickham. The generalized pairs plot. *Journal of Computational and Graphical Statistics*, 22(1):79–91, 2013.

[27] Elena Fanea, Sheelagh Carpendale, and Tobias Isenberg. An interactive 3d integration of parallel coordinates and star glyphs. In *Proc.*, INFOVIS'05, pages 20–28, 2005. ISBN 0-7803-9464-x.

[28] William H. Fletcher. Phrases in english, . URL http://www.phrasesinenglish.org/.

[29] William H. Fletcher. Web as corpus, . URL http://www.webascorpus.org/.

[30] Michael Friendly. A brief history of the mosaic display. *Journal of Computational and Graphical Statistics*, 11:89–107, 2001.

[31] Chris Gerrard. Tableau colors. http://public.tableausoftware.com/profile/chris.gerrard#!/vizhome/TableauColors/ColorPaletteswithRGBValues. [Online; accessed 2015-02-13].

[32] Bela Gipp, Norman Meuschke, Corinna Breitinger, Mario Lipinski, and Andreas Nuernberger. Demonstration of Citation Pattern Analysis for Plagiarism Detection. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Dublin, UK, Jul. 28 - Aug. 1 2013. ACM.

[33] GoogleLabs. Google scribe. URL http://scribe.googlelabs.com/.

[34] D. Gotz and H. Stavropoulos. Decisionflow: Visual analytics for high-dimensional temporal event sequence data. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1783–1792, Dec 2014.

[35] Martin Graham and Jessie Kennedy. Using curves to enhance parallel coordinate visualisations. In *Proc.of the Seventh International Conference on Information Visualization*, pages 10–16, 2003. ISBN 0-7695-1988-1. URL http://dl.acm.org/citation.cfm?id=938981.939625.

[36] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. Lineup: Visual analysis of multi-attribute rankings. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2277–2286, Dec 2013.

[37] ISOVIS group. Text Visualization Browser – A Visual Survey of Text Visualization Techniques. http://textvis.lnu.se, 2015. [Online; accessed 2015-04-14].

[38] Matthias Hagen, Martin Potthast, Benno Stein, and Christof Bräutigam. Query Segmentation Revisited. In Sadagopan Srinivasan, Krithi Ramamritham, Arun Kumar, M. P. Ravindra, Elisa Bertino, and Ravi Kumar, editors, *20th International Conference on World Wide Web (WWW 11)*, pages 97–106. ACM, March 2011.

[39] Sven Ove Hansson. Decision theory: A brief introduction. http://people.kth.se/~soh/decisiontheory.pdf, 2014. [Online; accessed 2014-02-14].

[40] Tam Harbert. The rise of the dataviz expert. http://www.networkworld.com/article/2166570/application-performance-management/tech-hotshots--the-rise-of-the-dataviz-expert.html, 2013. [Online; accessed 2014-02-14].

[41] Chris Harrsion. Web trigrams. URL http://www.chrisharrison.net/projects/visualization.html.

[42] S. Havre, B. Hetzler, and L. Nowell. Themeriver: visualizing theme changes over time. In *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*, pages 115–123, 2000.

[43] Jeffrey Heer and Stuart K. Card. Doitrees revisited: scalable, space-constrained visualization of hierarchical data. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 421–424. ACM, 2004. ISBN 1-58113-867-9.

[44] Julian Heinrich and Daniel Weiskopf. Continuous parallel coordinates. *IEEE TVCG*, 15:1531–1538, November 2009. ISSN 1077-2626.

[45] Tara Holland (SAS). SAS DATA VISUALIZATION & ANALYTIC DECISION MAKING, 2013. URL http://www.sas.com/offices/NA/canada/downloads/UserGroups/Regina-May2013/Holland-DataVisualization.pdf.

[46] Danny Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE TVCG*, 12:741–748, September 2006. ISSN 1077-2626.

[47] Werner Horn, Christian Popow, Lukas Unterasinger, et al. Support for fast comprehension of icu data: Visualization using metaphor graphics. *Methods Inf Med*, 40(5):421–424, 2001.

[48] Graham Hughes. How big is 'big data' in healthcare? http://blogs.sas.com/content/hls/2011/10/21/how-big-is-big-data-in-healthcare/, 2011. [Online; accessed 2015-04-28].

[49] Jean-Francois Im, Michael J. McGuffin, and Rock Leung. Gplom: The generalized plot matrix for visualizing multidimensional multivariate data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2606–2614, 2013. ISSN 1077-2626.

[50] Alfred Inselberg. *Parallel Coordinates, Visual Multidimensional Geometry and Its Applications*. Springer, 2009. ISBN 978-0-387-21507-5.

[51] iParadigms LLC. TurnItIn. http://turnitin.com/, 2014. [Online; accessed 1-December-2014].

[52] Stefan Jänicke, Annette Geßner, Marco Büchler, and Gerik Scheuermann. Visualizations for text re-use. In *IVAPP 14: Proceedings of the 5th International Conference on Information Visualization Theory and Application*. SCITEPRESS, SCITEPRESS, 2014.

[53] Sean Kandel, Andreas Paepcke, Joseph M Hellerstein, and Jeffrey Heer. Enterprise data analysis and visualization: An interview study. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2917–2926, 2012.

[54] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melancon. Visual analytics: Definition, process, and challenges. In Andreas Kerren, John T. Stasko, Jean-Daniel Fekete, and Chris North, editors, *Information Visualization*, volume 4950 of *Lecture Notes in Computer Science*, pages 154–175. Springer Berlin Heidelberg, 2008.

[55] Stefanie Klum, Petra Isenberg, Ricardo Langner, Jean-Daniel Fekete, and Raimund Dachselt. Stackables: combining tangibles for faceted browsing. In *International Working Conference on Advanced Visual Interfaces, AVI '12, Capri Island, Naples, Italy, May 22-25, 2012, Proceedings*, pages 241–248, 2012.

[56] Frank H. Knight. *Risk Uncertainty and Profit*. Martino Fine Books, 1921.

[57] S. Koch, M. John, M. Worner, A. Muller, and T. Ertl. Varifocalreader – in-depth visual analysis of large text documents. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1723–1732, Dec 2014. ISSN 1077-2626.

[58] Artem Konev, Jürgen Waser, Bernhard Sadransky, Daniel Cornel, Rui A. P. Perdigão, Zsolt Horváth, and M. Eduard Gröller. Run watchers: Automatic simulation-based decision support in flood management. *IEEE Trans. Vis. Comput. Graph.*, 20(12):1873–1882, 2014.

[59] Robert Kosara, Fabian Bendix, and Helwig Hauser. Parallel sets: Interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):558–568, July 2006. ISSN 1077-2626.

[60] Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. *Automated Grammatical Error Detection for Language Learners*. Morgan and Claypool Publishers, 2010. ISBN 1608454703, 9781608454709.

[61] Alexander Lex, Marc Streit, Christian Partl, Karl Kashofer, and Dieter Schmalstieg. Comparative analysis of multidimensional, quantitative data. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization / Information Visualization 2010)*, 16(6):1027–1035, 2010.

[62] K. Madhavan, N. Elmqvist, M. Vorvoreanu, Xin Chen, Yuetling Wong, Hanjun Xian, Zhihua Dong, and A. Johri. Dia2: Web-based cyberinfrastructure for visual analysis of funding portfolios. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1823–1832, Dec 2014. ISSN 1077-2626.

[63] A. Malik, R. Maciejewski, S. Towers, S. McCullough, and D.S. Ebert. Proactive spatiotemporal resource allocation and predictive visual analytics for community policing and law enforcement. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1863–1872, Dec 2014.

[64] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.

[65] K. T. McDonnell and K. Mueller. Illustrative parallel coordinates. *Computer-Graphics Forum*, (27):1031–1038, 2008.

[66] Norman Meuschke and Bela Gipp. State of the Art in Detecting Academic Plagiarism. *International Journal for Educational Integrity*, 9(1):50–71, Jun. 2013.

[67] Jean-Baptiste Michel, Yuan K. Shen, Aviva P. Aiden, Adrian Veres, Matthew K. Gray, The Google Books Team, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez L. Aiden. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, January 14 2011.

[68] Microsoft Support. How to Use the Windiff.exe Utility. http://support.microsoft.com/KB/159214, 2014. [Online; accessed 2014-12-01].

[69] Henry Mintzberg, Duru Raisinghani, and Andre Theoret. The structure of unstructured decision processes. *Administrative Science Quarterly*, 21:246–275, 1976.

[70] Misc. Anonymus Authors. GuttenPlag - kollaborative Plagiatsdokumentation. http://de.guttenplag.wikia.com/wiki/GuttenPlag_Wiki, 2014. [Online; accessed 1-December-2014].

[71] Misc. Anonymus Authors. VroniPlag Wiki - kollaborative Plagiatsdokumentation (Eine kritische Auseinandersetzung mit Hochschulschriften). http://de.vroniplag.wikia.com/wiki/Home, 2014. [Online; accessed 1-December-2014].

[72] Misc. Authors of Wikipedia. Alfred the Great. http://en.wikipedia.org/wiki/Alfred_the_Great, 2014. [Online; accessed 1-December-2014],.

[73] Tamara Munzner. *Visualization Analysis and Design (AK Peters Visualization Series)*. A K Peters/CRC Press, 2014. ISBN 1466508914.

[74] Matej Novotny and Helwig Hauser. Outlier preserving focus+context visualization in parallel coordinates. *IEEE TVCG*, (12):893–900, 2006.

[75] OpenDataCity (Datenfreunde UG) and Verein europe-v-facebook.org. Lobbyplag.eu. http://lobbyplag.eu/, 2014. [Online; accessed 1-December-2014].

[76] W. Bradford Paley. Textarc: Showing word frequency and distribution in text. URL http://www.textarc.org/appearances/InfoVis02/InfoVis02_TextArc.pdf. Poster Infovis 2002.

[77] Taehyun Park, Edward Lank, Pascal Poupart, and Michael Terry. Is the sky pure today? awkchecker: an assistive tool for detecting and correcting collocation errors. In *UIST '08: Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 121–130. ACM, 2008. ISBN 978-1-59593-975-3.

[78] Wei Peng, Matthew O. Ward, and Elke A. Rundensteiner. Clutter reduction in multi-dimensional data visualization using dimension reordering. In *Proc.*, INFOVIS'04, pages 89–96, 2004. ISBN 0-7803-8779-3.

[79] Charles Perin, Romain Vuillemot, and Jean-Daniel Fekete. Soccerstories: A kick-off for visual soccer analysis. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2506–2515, December 2013. ISSN 1077-2626.

[80] Catherine Plaisant, Brett Milash, Anne Rose, Seth Widoff, and Ben Shneiderman. Lifelines: Visualizing personal histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '96, pages 221–227, New York, NY, USA, 1996. ACM. ISBN 0-89791-777-4.

[81] Catherine Plaisant, Jesse Grosjean, and Benjamin B. Bederson. Spacetree: Supporting exploration in large node link tree, design evolution and empirical

evaluation. In *INFOVIS '02: Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, page 57, Washington, DC, USA, 2002. IEEE Computer Society. ISBN 0-7695-1751-X.

[82] Martin Potthast. Picapica. http://http://www.picapica.org. [Online; accessed 2015-02-13].

[83] Martin Potthast, Benno Stein, Paolo Rosso, and Efstathios Stamatatos. PAN Website. http://pan.webis.de. [Online; accessed 2015-02-14].

[84] Martin Potthast, Matthias Hagen, Michael Völske, and Benno Stein. Crowdsourcing Interaction Logs to Understand Text Reuse from the Web. In Pascale Fung and Massimo Poesio, editors, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 13)*, pages 1212–1221. ACL, August 2013. URL http://www.aclweb.org/anthology/P13-1119.

[85] Martin Potthast, Matthias Hagen, Anna Beyer, Matthias Busse, Martin Tippmann, Paolo Rosso, and Benno Stein. Overview of the 6th International Competition on Plagiarism Detection. In Linda Cappellato, Nicola Ferro, Martin Halvey, and Wessel Kraaij, editors, *Working Notes Papers of the CLEF 2014 Evaluation Labs*, CEUR Workshop Proceedings. CLEF and CEUR-WS.org, September 2014.

[86] Daniel Power. *Decision support basics*. Business Expert Press, New York, N.Y. (222 East 46th Street, New York, NY 10017, 2009. ISBN 1606490834. URL http://ebooks.businessexpertpress.com/Books/9781606490839/f4670371-4b9d-4244-ace0-e9fa90fcf509.

[87] Prio Infocenter AB. Urkund. http://www.urkund.com, 2015. [Online; accessed 5-February-2015].

[88] Research and Development Unit for English Studies. Webcorp live. URL http://www.webcorp.org.uk/.

[89] Philip Resnik and Aaron Elkiss. The linguist's search engine: an overview. In *ACL '05: Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, pages 33–36, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[90] Patrick Riehmann, Manfred Hanfler, and Bernd Froehlich. Interactive sankey diagrams. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS '05, pages 31–. IEEE Computer Society, 2005. ISBN 0-7803-9464-x.

[91] Patrick Riehmann, Henning Gruendl, Bernd Froehlich, Martin Potthast, Martin Trenkmann, and Benno Stein. The Netspeak WordGraph: Visualizing Keywords in Context. In Giuseppe Di Battista, Jean-Daniel Fekete, and Huamin

Qu, editors, *4th IEEE Pacific Visualization Symposium (PacificVis 11)*, pages 123–130. IEEE, March 2011.

[92] Patrick Riehmann, Wieland Möbus, and Bernd Froehlich. Visualizing food ingredients for children by utilizing glyph-based characters. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, AVI '14, pages 133–136, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2775-6.

[93] Alexander Rind, S. Miksch, Wolfgang Aigner, Thomas Turic, and Margit Pohl. Visuexplore: Gaining new medical insights from visual exploration. In Gillian R Hayes and Desney S Tan, editors, *Proceedings of the 1st International Workshop on Interactive Systems in Healthcare (WISH@CHI2010)*, pages 149–152, 2010. ISBN 9780982628485. URL http://publik.tuwien.ac.at/files/PubDat_190298.pdf.

[94] Alexander Rind, Taowei David Wang, Wolfgang Aigner, S. Miksch, Krist Wongsuphasawat, Catherine Plaisant, and Ben Shneiderman. Interactive information visualization to explore and query electronic health records. *Foundations and Trends in Human-Computer Interaction*, 5:207–298, 02/2013 2013. URL http://publik.tuwien.ac.at/files/PubDat_214284.pdf.

[95] Scenario. Project scene graph [online]. Available: https://scenegraph.dev.java.net/.

[96] SciPlore. CitePlag demonstrates Citation-based Plagiarism Detection (CbPD). http://citeplag.org/ and http://sciplore.org/, 2014. [Online; accessed 1-December-2014],.

[97] Conglei Shi, Weiwei Cui, Shixia Liu, Panpan Xu, Wei Chen, and Huamin Qu. Rankexplorer: Visualization of ranking changes in large time series data. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2669–2678, Dec 2012.

[98] Ben Shneiderman. Dynamic queries for visual information seeking. *IEEE Software*, 11:70–77, 1994.

[99] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, VL '96, pages 336–, Washington, DC, USA, 1996. IEEE Computer Society. ISBN 0-8186-7508-X.

[100] Harri Siirtola. Direct manipulation of parallel coordinates. In *CHI '00 extended abstracts on Human factors in computing systems*, CHI EA '00, pages 119–120, 2000. ISBN 1-58113-248-4.

[101] Simon and Stuart M. Dillon. The new science of management decision. In *In Proceedings of the 33 rd Conference of the Operational Research Society of New Zealand*, 1960.

[102] Tableau Software. Tableau. http://www.tableau.com, 2015. [Online; accessed 2015-04-27].

[103] Robert Spence and Maureen Parr. Cognitive assessment of alternatives. *Interacting with Computers*, 3(3):270 – 282, 1991. ISSN 0953-5438.

[104] Robert Spence and Lisa Tweedie. The attribute explorer: information synthesis via exploration. interacting with. *Computers*, 11:137–146, 1998.

[105] David Stodder. TDWI Best Practices Report | Data Visualization and Discovery for Better Business Decisions, 2013. URL http://www.sas.com/offices/NA/canada/downloads/UserGroups/Regina-May2013/Holland-DataVisualization.pdf.

[106] Marc Streit, Alexander Lex, Samuel Gratzl, Christian Partl, Dieter Schmalstieg, Hanspeter Pfister, Peter J. Park, and Nils Gehlenborg. Guided visual exploration of genomic stratifications in cancer. *Nature Methods*, 11(9):884–885, September 2014. ISSN 1548-7091. URL http://www.nature.com/nmeth/journal/v11/n9/full/nmeth.3088.html.

[107] Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiko Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions On Systems, Man, And Cybernetics*, SMC-11(2):109–125, 1981.

[108] Bongwon Suh, Ed H. Chi, Aniket Kittur, and Bryan A. Pendleton. Lifting the veil: Improving accountability and social transparency in wikipedia with wikidashboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1037–1040, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-011-1.

[109] Holger Theisel. Higher order parallel coordinates. *Proc.of VMV '00*, pages 415–420, 2000. URL http://www.isg.cs.uni{}-magdeburg.de/visual/files/publications/Archive/Theisel{_}2000{_}VMAV.pdf.

[110] Martin Theus. Interactive data visualization using mondrian. *Journal of Statistical Software*, 7(11):1–9, 11 2002. ISSN 1548-7660. URL http://www.jstatsoft.org/v07/i11.

[111] Christian Tominski and Wolfgang Aigner. The TimeViz Browser – A Visual Survey of Visualization Techniques for Time-Oriented Data. http://survey.timeviz.net, 2015. [Online; accessed 2015-04-14].

[112] Lisa Tweedie, Bob Spence, Huw Dawkes, and Hua Su. The influence explorer. In *Conference Companion on Human Factors in Computing Systems*, CHI '95, pages 129–130. ACM, 1995. ISBN 0-89791-755-3.

[113] Alfred Inselberg und Bernard Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensinal geometry. *VIS '90 Proc.of the 1st conference on Visualization '90*, 1990.

[114] User8 (Pseudonym in Guttenplag Wiki). Herausragende Quellen. http://de.guttenplag.wikia.com/wiki/Visualisierungen and http://de.guttenplag.wikia.com/wiki/Herausragende_Quellen, 2014. [Online; accessed 1-December-2014].

[115] S. van den Elzen and J.J. van Wijk. Baobabview: Interactive construction and analysis of decision trees. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 151–160, Oct 2011.

[116] Frank van Ham, Martin Wattenberg, and Fernanda B. Viégas. Mapping text with phrase nets. *IEEE Transactions on Visualization and Computer Graphics*, 15(06):1169–1176, 2009. ISSN 1077-2626.

[117] Jarke J. van Wijk and Robert van Liere. Hyperslice - visualization of scalar functions of many variables, 1993.

[118] Fernanda Viegas and Martin Wattenberg. Web seer. URL http://hint.fm/projects/seer/.

[119] Fernanda B. Viégas, Martin Wattenberg, and Kushal Dave. Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 575–582, New York, NY, USA, 2004. ACM. ISBN 1-58113-702-8.

[120] Georg von Mayr. *Die Gesetzmässigkeit im Gesellschaftsleben: statistische Studien*. Oldenbourg, München, 1887.

[121] Yingxu Wang and Günther Ruhe. The cognitive process of decision making. *IJCINI*, 1(2):73–85, 2007.

[122] Martin Wattenberg. Baby names, visualization, and social data analysis. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS '05, pages 1–, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7803-9464-x.

[123] Martin Wattenberg and Fernanda B. Viégas. The word tree, an interactive visual concordance. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1221–1228, 2008. ISSN 1077-2626.

[124] Martin Wattenberg, Fernanda B. Viégas, and Katherine Hollenbach. Visualizing activity on wikipedia with chromograms. In *Proceedings of the 11th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part II*, INTERACT'07, pages 272–287, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-74799-0, 978-3-540-74799-4.

[125] Debora Weber-Wulff. Tests of plagiarism software. http://plagiat.htw-berlin.de/software-en/. [Online; accessed 2015-02-13].

[126] Debora Weber-Wulff. *False Feathers : a Perspective on Academic Plagiarism*. Springer Berlin, Berlin, 2014. ISBN 3642399606.

[127] Debora Weber-Wulff, Christopher Möer, Jannis Touras, and Elin Zincke. Plagiarism detection software test 2013. http://plagiat.htw-berlin.de/software-en/test2013/report-2013/. [Online; accessed 2015-03-01].

[128] Webis Group at Bauhaus-Universität Weimar. Netspeak writing assistance. URL http://netspeak.cc.

[129] Ryen W. White and Dan Morris. Investigating the querying and browsing behavior of advanced search engine users. In *Proceedings of the 30th ACM SIGIR conference*, pages 255–262, 2007.

[130] Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1129–1136, November 2007. ISSN 1077-2626.

[131] Christopher Williamson and Ben Shneiderman. The dynamic homefinder: Evaluating dynamic queries in a real-estate information exploration system. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '92, pages 338–346. ACM, 1992. ISBN 0-89791-523-2.

[132] Eberhard Witte. Field research on complex decision-making processes – the phase theorem, 1972.

[133] Kent Wittenburg, Tom Lanning, Michael Heinrichs, and Michael Stanton. Parallel bargrams for consumer-based information exploration and choice. In *UIST*, pages 51–60, 2001.

[134] Nelson Wong, Sheelagh Carpendale, and Saul Greenberg. Edgelens: an interactive method for managing edge congestion in graphs. In *Proc.*, INFOVIS'03, pages 51–58, 2003. ISBN 0-7803-8154-8. URL http://dl.acm.org/citation.cfm?id=1947368.1947382.

[135] Jing Yang, Wei Peng, Matthew O. Ward, and Elke A. Rundensteiner. Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In *Proc.*, INFOVIS'03, pages 105–112, 2003. ISBN 0-7803-8154-8. URL http://dl.acm.org/citation.cfm?id=1947368.1947390.

[136] Ji Soo Yi. *Visualized decision making: development and application of information visualization techniques to improve decision quality of nursing home choice*. Doctoral thesis, Georgia Institute of Technology, 2008. URL https://smartech.gatech.edu/bitstream/handle/1853/24662/yi_jisoo_200808_phd.pdf.

[137] Xiaoru Yuan, Peihong Guo, He Xiao, Hong Zhou, and Huamin Qu. Scattering points in parallel coordinates. *IEEE TVCG*, 15:1001–1008, November 2009. ISSN 1077-2626.

LIST OF FIGURES

LIST OF TABLES

EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Teile der Arbeit, die bereits Gegenstand von Prüfungsarbeiten waren, sind ebenfalls unmissverständlich gekennzeichnet. Bei der Auswahl und Auswertung folgenden Materials haben mich die nachstehend aufgeführten Personen in der jeweils beschriebenen Weise unentgeltlich unterstützt:

1. (von mir betreut) Henning Gründl bei der Programmierung einer frühen Version des Wordgraph im Rahmen seiner deutschsprachigen Bachelorarbeit “Wordgraph” und bei der gemeinsamen Erstellung der Bilder 2, 3, 5, 8,11 und 12
2. (von mir betreut) Jens Opolka bei der Programmierung einer frühen Version des Product Explorer im Rahmen seiner deutschsprachigen Bachelorarbeit “Parallel Queries”.
3. (von Dr. Martin Potthast und mir betreut) Maximilian Michel and Jan Grassegger bei der Programmierung einer frühen Version der Plagiatsvisualisierung.
4. (von Dr. Martin Potthast und mir betreut) Stefanie Wetzel, Dora Spensberger, and Christof Bräutigam für die Akquise, Säuberung und Aufbereitung der Daten von VroniPlag [71] und GuttenPlag [70]
5. Dr. Martin Potthast und Prof. Benno Stein verantwortlich für Section 2.6 und für Abbildung 13
6. Dr. Martin Potthast, Prof. Benno Stein und Prof. Bernd Fröhlich für wissenschaftliche und formale Korrektur
7. Loren O’Dell für die Englischkorrektur

Weitere Personen waren an der inhaltlich-materiellen Erstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs- bzw. Beratungsdiensten (Promotionsberater oder anderer Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt. Ich versichere, dass ich nach bestem Wissen die reine Wahrheit gesagt und nichts verschwiegen habe.

Gotha, Mai 2015

Patrick Riehmann

COLOPHON

This document was typeset using a version of the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

http://code.google.com/p/classicthesis/

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

http://postcards.miede.de/

Final Version as of September 7, 2015 (`classicthesis` version 4.1).